



UNIVERSIDAD
CARLOS III DE MADRID

BACHELOR THESIS

Analysis and Implementation of Gravity Field Models For Planets and Asteroids

Author:

Diego García Pardo

Tutor:

Manuel Sanjurjo Rivo

Director:

Daniel González Arribas

June 21, 2014

Contents

I	INTRODUCTION TO GRAVITY MODELS	7
II	SPHERICAL HARMONICS SOLUTION OF GRAVITATIONAL POTENTIAL FUNCTION	10
II.1	Introduction	11
II.2	Aspherical Potential Function	12
II.2.1	Spherical Harmonics Potential Function	13
II.2.1.1	Normalizing Function	14
II.2.1.2	Potential Function Convergence	17
II.2.1.3	Potential Energy Variation At Constant Altitude . .	19
II.3	Orbit Propagation	20
II.3.1	Computing The Acceleration Field	20
II.3.1.1	Recursive Acceleration	21
II.3.1.2	Acceleration 3D Vector Component	21
II.3.1.3	Non-Inertial Acceleration	24
II.3.2	Algorithm Implementation	25
II.3.3	Integrator, Trajectory Computation	26
II.3.3.1	Trajectory Integration Validation in MatLab	27
III	DISCRETE BODIES GRAVITATION	30
III.1	Introduction	31
III.2	Discrete Body Shape Determination	32
III.3	Exterior Gravitation Algorithm	34
III.3.1	Surface Integral Over a Polygonal Surface	35
III.3.2	Common Edges, Algorithm Optimization	39

III.3.3	Gradient of Potential Function Acceleration	40
III.3.4	Summary of Results	41
IV	SPHERICAL HARMONICS OF A CONSTANT DENSITY POLYHEDRON	42
IV.1	Spherical Harmonics Mass Coefficients	43
IV.2	Standard Simplex Integration	47
IV.2.1	Hypothesis	47
IV.2.2	Change Of Coordinates	47
IV.2.3	Integral Solution	50
IV.2.4	Algorithm Implementation Comments	51
V	APPLICATIONS	52
V.1	Earth Gravity Field And WGS84 Ellipsoid	53
V.1.1	Gravity Intensity Variation	54
V.1.2	WGS84 Ellipsoid Gravity Field	55
V.1.2.1	The Mass Constraint	55
V.1.2.2	Spherical Harmonics Mass Coefficients of Discrete Body	57
V.1.2.3	Evaluation of Gravity Field	60
V.2	Asteroid Castalia Analysis	64
V.2.1	Gravity Field Intensity Near Surface	65
V.2.2	Methods Comparison	69
VI	CONCLUSIONS	71
A	ANNEX, Software Manual	75
A.1	Implementation in C	76
A.1.1	Executable Files User Manual	77
A.1.1.1	Executable File, Simple Calculations	79
A.2	Implementation in MatLab	80

A.2.1	Part II,Spherical Harmonics	80
A.2.2	Part III, Discrete Body Gravitation	81
A.2.3	Part IV, Spherical Harmonics Mass Coefficients of a Con- stant Density Polyhedron	81

B	ANNEX, Economical Cost	84
----------	-------------------------------	-----------

ABSTRACT

ABSTRACT

Gravitational models are an essential tool to develop, simulate and predict the behaviour of satellites, asteroids and other bodies of celestial nature. The accuracy of these models is of critical importance in surveillance, orbit determination and propagation.

In the following pages, it will be presented two different models to study the effect of gravity fields produced by both Planets and Asteroids.

The reader must always note the importance of improving the results obtained by means of classical gravitational theories as modern society dependence in man-made space satellites is constantly growing, implying a direct application in the industrial and academic fields. For example, Rummel ([7]) expects that the data retrieved by GRACE and GOCE missions will help to model with even higher precision the gravity levels as well as time variations due to tidal effects (both fluid and solid tides). The models here presented try to average this effects and to predict accurately the gravity vector field near Earth but always neglecting time variations of the gravitational field.

These models have been prototyped in MatLab and implemented in C programming languages. MatLab may be used for preliminary results as well as for data analysis and interpretation. Mathematically speaking, these models cover wide numerical ranges of magnitude that can lead to overflow or underflow. Not in vane, big effort has been devoted to normalize the magnitude of the data used in calculations as well as special care in function evaluations. Also, MatLab parallel toolbox allows to perform easily parallel calculation without extra-effort.

The implementation of such algorithms has already been a matter of concern as the algorithms have to be evaluated as fast as possible. The polyhedron algorithm that will be presented later can be highly parallelized and hardware may make the difference rather than software. For example in [2] it can be found a performance analysis comparing single-threaded implementation with massively parallelized algorithm in CUDA GPU.

The theories presented are based on classical Newton's laws but are intended to improve the predictions of Newton's law of universal gravitation. However it is not the scope of this report to introduce relativistic effects as the variations in the results as it will be shown are below the uncertainty level of

the predictions done with the available data. Neither atmospheric drag nor solar radiation pressure have been modelled as they depend on the shape of the body. Only gravitational forces are studied.

Part I

INTRODUCTION TO GRAVITY MODELS

Introduction To Gravity

In [11] and in [6], main sources of information of this work, it is possible to find solutions to the gravitational potential function in the framework of Newtonian mechanics. The rigorous way to describe the gravitational field of a massive body is simply by means of summing the contributions of the individual mass elements in the body. The gravitational acceleration can be then expressed as:

$$\vec{g} = G \int_{\text{Body}} \frac{\rho(\vec{s}) (\vec{r} - \vec{s}) d^3\vec{s}}{|\vec{r} - \vec{s}|^3} \quad (\text{I.1})$$

Where \vec{g} is the acceleration vector, G and M are the universal gravitational constant and the central body mass. ρ is the central body density and \vec{s} is the vector pointing to the mass elements. \vec{r} is the field point, the space location where the algorithm pretends to evaluate the gravitational field.

The previous result is one of the main achievements of Newtonian mechanics. This definition of gravity happens to produce a conservative force field, and mathematically, the problem is equivalent to solve the following one:

$$\nabla U = \vec{g} \quad (\text{I.2})$$

The approach from both methods described in this work is to solve for the potential function employing the integral shape of equation I.2

$$U = G \int_{\text{Body}} \frac{\rho(\vec{s}) d^3\vec{s}}{|\vec{r} - \vec{s}|} \quad (\text{I.3})$$

From this point, both Spherical Harmonics and Polyhedron models depart. Both methods solve the integral Eq. I.1 in a exact analytical way **under the assumption of constant density**.

Summarizing, on one hand, Spherical Harmonics makes use of infinite series to solve the potential function in spherical coordinates by using associated Legendre functions ("ALF's"). On the other hand, polyhedron algorithm describes the gravitational field about a discretized massive body. This last algorithm solves the potential function equations thus leading to an analytical solution to the problem.

Accuracy of both methods only depends on the validity of the assumption of constant density as well as on the quality of the surface discretization [11]. More information on the methods used to retrieve shape data of an asteroid is given in part III

Note that the solution given in terms of spherical harmonics is used to described small deviation from a perfect spherical body as can occur with planets. The discrete body method, although being more general, its application may be directed to the evaluation of asteroid gravitational fields. In addition, planetary gravitation will be discussed using this method. It will be seen that the solution given by means of the so called "Spherical Harmonics" depends on "spherical harmonics mass coefficients". It will be seen that for the case of Earth, they can be directly obtained from agencies such as NASA or ESA, but it will be shown how to obtain them from a discretized surface.

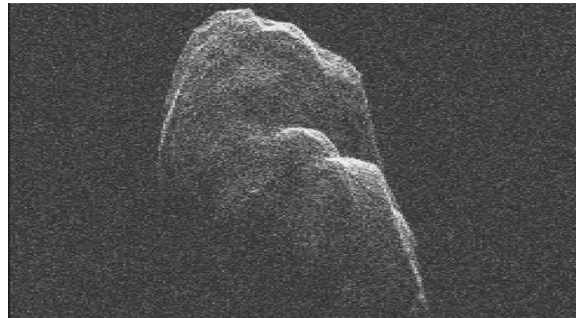


Figure I.0.1: Radar Image from Asteroid Toutatis. Source: NASA/JPL-Caltech

Part II

SPHERICAL HARMONICS SOLUTION OF GRAVITATIONAL POTENTIAL FUNCTION

Chapter II.1

Introduction

This method solves the potential function as expressed in Eq. (I.1) for a general body including all perturbations due to a non-perfectly spherical body. The potential function is then usually called “Aspherical Potential function”. This method although can be applied to a general arbitrary shaped body, the convergence is only guaranteed in the region outside of the so called: “*Reference Sphere*”, [11]. The reference sphere is defined to be the smallest sphere that being centred at the origin of the reference frame (which will always be the mass centre or centroid as the density is constant) circumscribes the all the points of the body. . This fact makes this method of primary application to planets, and above all, Earth gravitation.

From this point on, the spherical harmonics mass coefficients used are those determined by empirical methods. These are data retrieved from satellite observations which can be found in the databases. The reference sphere radius used to measure these coefficients is given and therefore it must be employed to be coherence with the method.

Lastly, it will be assumed from now on that density is constant and that there are no time variations of gravity as these depend on Earth solid tides and other phenomena of complex nature and whose model may vary the value of acceleration at most of 10^{-5} m/s^2 as referred in [6]).

Chapter II.2

Aspherical Potential Function

In the following pages, it will be presented the way to solve the gravitational potential function. In [6], this is referred to as the "perturbed potential function" since it takes into account small deviations (perturbations) that appear from the fact the in nature is very improbable to find a perfect spherical body. This is the case for most of the planets, and in particular for Earth. This is where the term aspherical comes from.

In all solutions, it is a primary concern the definition of a coordinate system. Due to the lack of convention, the spherical coordinate system for this work will be defined as:

$$x = r \cos(\phi) \cos(\lambda) \tag{II.1}$$

$$y = r \cos(\phi) \sin(\lambda) \tag{II.2}$$

$$z = r \sin(\phi) \tag{II.3}$$

Therefore latitude, ϕ is measured from an x-y plane and longitude, λ is measured from the x-axis being positive towards East.

The x,y and z axes are Earth Fixed axis oriented such that ϕ and λ correspond to the *terrestrial* latitude and longitude.

II.2.1 Spherical Harmonics Potential Function

Recalling our departure point :

$$U(r, \phi, \lambda) = G \int_{\text{Body}} \frac{\rho(\vec{s}) d^3 \vec{s}}{|\vec{r} - \vec{s}|}$$

The vector \vec{r} defines the location in space where the potential function is to be evaluated and the vector \vec{s} defines the points in space with mass which contribute to the value of the potential. The previous integral equation can be expanded in terms of Legendre Polynomials, or more generally, the associated Legendre functions $P_{n,m}$ ¹. Note that γ is the angle between vectors \vec{r} and \vec{s}

$$\frac{1}{|\vec{r} - \vec{s}|} = \frac{1}{r} \sum_{n=0}^{\infty} \left(\frac{s}{r}\right)^n P_n(\cos(\gamma)) \quad (\text{II.4})$$

$$P_{n,m}(\nu) = (1 - \nu^2)^{m/2} \frac{d^m}{d\nu^m} P_n(\nu) \quad (\text{II.5})$$

Making use of the addition theorem of Legendre Polynomials, it is possible to express the potential function as:

$$U = \frac{GM}{r} \sum_{n=0}^{\infty} \sum_{m=0}^n \left(\frac{R}{r}\right)^n P_{nm}(\sin(\phi))(C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)) \quad (\text{II.6})$$

Where the coefficients $C_{n,m}$ & $S_{n,m}$ are intrinsically related to mass distribution inside the body, they are the so called "spherical harmonics mass coefficients".

These coefficients can be found via analytical methods if the density distribution of the body as well as its shape were perfectly known. As this is not usually the case, these coefficients are evaluated the other way around from satellite measurements. Later in this paper it will be presented how to obtain the spherical harmonics of a discretized volume body. These constants

¹This is only valid for $|r| > |s|$ which is the region of space of interest.

usually cover wide ranges of magnitude making numerical evaluation to be limited by round-off errors, problem that can be handled by normalizing the coefficient values. This normalization process smoothes the magnitudes of the coefficients. An empirical estimation found in [6] can be given by *Kaula rule*:

$$\bar{C}_{n,m}, \bar{S}_{n,m} \approx \frac{10^{-5}}{n^2} \quad (\text{II.7})$$

where $\bar{C}_{n,m}, \bar{S}_{n,m}$ are the normalized spherical harmonics mass coefficients $C_{n,m}, S_{n,m}$

II.2.1.1 Normalizing Function

In order to normalize the range of magnitude of such coefficients (also the value of the Legendre polynomial), the following normalizing function is defined:

$$N_{n,m} = \sqrt{\frac{(2 - \delta_{0,m})(2n+1)(n-m)!}{(n+m)!}} \quad (\text{II.8})$$

The normalized coefficients are defined as in [6] :

$$\bar{P}_{n,m} = N_{n,m} P_{n,m} \quad (\text{II.9})$$

$$\bar{C}_{n,m} = N_{n,m}^{-1} C_{n,m} \quad (\text{II.10})$$

$$\bar{S}_{n,m} = N_{n,m}^{-1} S_{n,m} \quad (\text{II.11})$$

Finally the potential function expressed in terms of normalized coefficients is:

$$U = \frac{GM}{r} \sum_{n=0}^{\infty} \sum_{m=0}^n \left(\frac{R}{r}\right)^n \bar{P}_{nm}(\sin(\phi)) (\bar{C}_{nm} \cos(m\lambda) + \bar{S}_{nm} \sin(m\lambda)) \quad (\text{II.12})$$

Normalized Associated Legendre Polynomial

Below the recursive scheme used to obtain first the associated legendre polynomials, and later the normalized version of them is presented. A recursive scheme is used to increase the speed of computations and to reduce numerical round-off errors. This increase in accuracy is possible since normalization is done at each recursion step.

$$P_{00}(\nu) = 1 \text{ (seed)} \quad (\text{II.13})$$

$$P_{m,m}(\nu) = (2m-1) (1-\nu^2)^{\frac{1}{2}} P_{m-1,m-1} \quad (\text{II.14})$$

$$P_{m+1,m}(\nu) = (2m+1) \nu P_{m,m} \quad (\text{II.15})$$

$$P_{n,m}(\nu) = \frac{1}{n-m} [(2n-1)\nu P_{n-1,m} - (n+m-1)P_{n-2,m}] \quad (\text{II.16})$$

To produce a normalized recursion it is a must to take into account a new factor appearing when different order and degree ALF's are included in the same equation.

For example in II.16 in order to transform the recursive scheme to obtain $\bar{P}_{n,m}$ the equation would be left as:

$$\begin{aligned} \bar{P}_{n,m}(\nu) = \frac{1}{n-m} \left[(2n-1) \nu \frac{N_{n-1,m}}{N_{n,m}} \bar{P}_{n-1,m} \right. \\ \left. - (n+m-1) \frac{N_{n-2,m}}{N_{n,m}} \bar{P}_{n-2,m} \right] \end{aligned} \quad (\text{II.17})$$

Also it happens that it is possible to cancel factorials and left the new coefficients as function of basic arithmetic operations such as square root, multiplication or division.

The same result is reached in [8], So finally the normalized recursive scheme is:

$$\bar{P}_{0,0} = 1 \quad (II.18)$$

$$\bar{P}_{1,1} = (1 - \nu^2)^{\frac{1}{2}} \sqrt{3} \bar{P}_{0,0} \quad (II.19)$$

$$\bar{P}_{m,m} = (1 - \nu^2)^{\frac{1}{2}} \sqrt{\frac{2m+1}{2m}} \bar{P}_{m-1,m-1} \text{ if and only if } m > 1 \quad (II.20)$$

$$\bar{P}_{m,m-1} = \nu \sqrt{2m+1} \bar{P}_{m-1,m-1} \quad (II.21)$$

$$\bar{P}_{n,m} = \frac{1}{n-m} \left[(2n-1) \nu \xi_1 \bar{P}_{n-1,m} - (n+m-1) \xi_2 \bar{P}_{n-2,m} \right] \quad (II.22)$$

Where in Eq. II.22

$$\xi_1 = \sqrt{\frac{2n+1}{2n-1} \frac{n-m}{n+m}} \quad (II.23)$$

$$\xi_2 = \sqrt{\frac{2n+1}{2n-3} \frac{n-m}{n+m} \frac{n-m-1}{n+m-1}} \quad (II.24)$$

The reader must also note that equation II.19 is an exception of the general case expressed in equation II.20 due to the factor $(2 - \delta_{0,m-1})$

II.2.1.2 Potential Function Convergence

The convergence of the model can be tested by increasing the size of the model that is being used and checking the value of the potential function. In this case the source of the value of $C_{n,m}$ & $S_{n,m}$ coefficients is from the NASA mission ITG Goce (year 2013) with a model of degree 240 spherical harmonics coefficients for planet Earth.

In order to evaluate the convergence of the aspherical potential function, it is possible to start by choosing an arbitrary single point in space and show graphically the evolution of the following expression:

$$U_k = \|U^k - U^{k-1}\| \quad (\text{II.25})$$

Where U^k represents the value of the potential function using a model of degree k i.e:

$$U^k = \frac{GM}{r} \sum_{n=0}^k \sum_{m=0}^n \left(\frac{R}{r}\right)^n \bar{P}_{nm}(\sin(\phi)) (\bar{C}_{nm} \cos(m\lambda) + \bar{S}_{nm} \sin(m\lambda)) \quad (\text{II.26})$$

The following plot shows the given result at an arbitrary location in space (Latitude And Longitude expressed in [degrees]):

It is possible to see in the previous figure that the results do converge as the difference for increasing model size between the values is each time smaller.

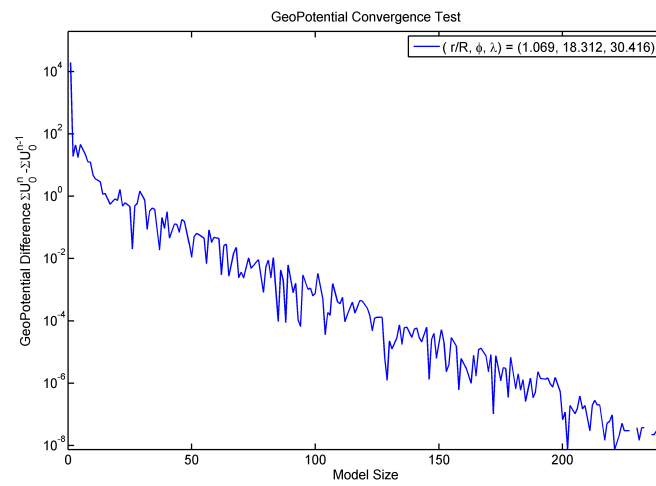


Figure II.2.1: Convergence Test on Gravitational Potential Function

II.2.1.3 Potential Energy Variation At Constant Altitude

As general trend the further from Earth, the faster the convergence (this occurs as the Earth is seen more and more as a point mass). However, the closer to Earth, the stronger the perturbations derived from a non-spherical body. The deviations from a perfect spherical body occur mainly because the equatorial diameter is higher than the polar diameter. Thus there is more mass near the equator. Longitudinal variation in gravity is merely due to non-uniformities in Earth mass distribution and can be modelled with higher accuracy in time dependent models as indicates [10]. Variations in latitude are mainly due to asphericity, with a contribution much higher than non-uniformities. This is also found in literature as the J_2 perturbation which is the highest found on planet Earth.

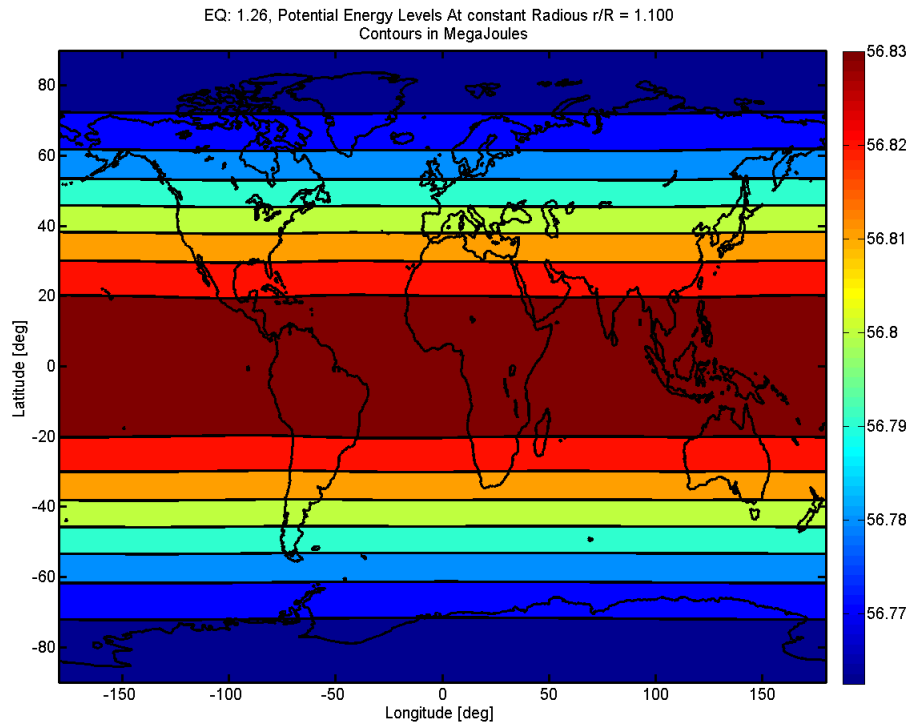


Figure II.2.2: Contours Of Gravitational Potential Energy

Chapter II.3

Orbit Propagation

II.3.1 Computing The Acceleration Field

Being able to compute the potential function, the solution for the acceleration can be obtained using a similar recursive scheme. The acceleration vector is defined as in eq. I.2

$$\vec{g} = \nabla U = \frac{dU}{dx}\vec{i} + \frac{dU}{dy}\vec{j} + \frac{dU}{dz}\vec{k} \equiv \ddot{x}\vec{i} + \ddot{y}\vec{j} + \ddot{z}\vec{k} \quad (\text{II.1})$$

This analytical expansion of the gradient of U is long and tedious. The equations developed by Cunningham (1970) which compute the acceleration (force field per unit mass) in terms of the GeoPotential are summarized in [6]

II.3.1.1 Recursive Acceleration

In order to get the acceleration components, first it is required to define 2 functions which will make clearer the expressions.

$$V_{n,m} = \left(\frac{R}{r}\right)^{n+1} P_{n,m}(\sin(\phi)) \cos(m\lambda) \quad (\text{II.2})$$

$$W_{n,m} = \left(\frac{R}{r}\right)^{n+1} P_{n,m}(\sin(\phi)) \sin(m\lambda) \quad (\text{II.3})$$

It must be noted, that the Pine's Algorithm outputs the normalized Legendre Polynomial, thus transformation to normalized functions is required.

$$\bar{V}_{n,m} = \left(\frac{R}{r}\right)^{n+1} \bar{P}_{n,m}(\sin(\phi)) \cos(m\lambda) \quad (\text{II.4})$$

$$\bar{W}_{n,m} = \left(\frac{R}{r}\right)^{n+1} \bar{P}_{n,m}(\sin(\phi)) \sin(m\lambda) \quad (\text{II.5})$$

II.3.1.2 Acceleration 3D Vector Component

In [6] one can find the recursive equations that develop x, y and z components of the acceleration. These are in terms of zonal ($m = 0$) and the other two coefficients (sectorial ($m = n$) and tesseral (rest)). The following equations express the solution in terms of non-normalize values.

–ZONAL TERMS RECURSIONS–

$$\ddot{x}_{n,0} = \frac{GM}{R^2} [-C_{n,0} V_{n+1,1}] \quad (\text{II.6})$$

$$\ddot{y}_{n,0} = \frac{GM}{R^2} [-C_{n,0} W_{n+1,1}] \quad (\text{II.7})$$

$$\ddot{z}_{n,0} = \frac{GM}{R^2} [(n+1)(-C_{n,0} V_{n+1,0}) - S_{n,0} W_{n+1,0}] \quad (\text{II.8})$$

–RESTING TERMS RECURSIONS–

$$\begin{aligned}\ddot{x}_{n,m} = & \frac{GM}{2R^2} [(-C_{n,m}V_{n+1,m+1} - S_{n,m}W_{n+1,m+1}) \\ & + \frac{(n-m+2)!}{(n+m)!} (C_{n,m}W_{n+1,m-1} + S_{n,m}V_{n+1,m-1})] \end{aligned} \quad (\text{II.9})$$

$$\begin{aligned}\ddot{y}_{n,m} = & \frac{GM}{2R^2} [(-C_{n,m}W_{n+1,m+1} + S_{n,m}V_{n+1,m+1}) \\ & + \frac{(n-m+2)!}{(n+m)!} (-C_{n,m}W_{n+1,m-1} + S_{n,m}V_{n+1,m-1})] \end{aligned} \quad (\text{II.10})$$

$$\ddot{z}_{n,m} = \frac{GM}{R^2} [(n-m+1)(-C_{n,m}V_{n+1,m}) - S_{n,m}W_{n+1,m}] \quad (\text{II.11})$$

Finally the acceleration vector at a given point is found as:

$$[g] = \nabla U = [a_x, a_y, a_z] = \left[\sum_{n,m} \ddot{x}_{n,m}, \sum_{n,m} \ddot{y}_{n,m}, \sum_{n,m} \ddot{z}_{n,m} \right] \quad (\text{II.12})$$

Normalizing Acceleration Recursion

As the coefficients are mainly found already normalized, the functions V and W are also preferably obtained in normalized shape, a correction factor must be applied as the coefficients and functions do not coincide in order and degree analogously to the process followed normalizing the recursive scheme of the Legendre polynomials.

$$C_{n,0}V_{n+1,1} \neq \bar{C}_{n,0}\bar{V}_{n+1,1}$$

This correction factor is to be applied as follows

$$C_{n,m}V_{u,v} = \bar{C}_{n,m}\bar{V}_{u,v} \Phi_{n,m}^{u,v} \quad (\text{And same applies for W}) \quad (\text{II.13})$$

$$\Phi_{n,m}^{u,v} = \frac{N_{n,m}}{N_{u,v}} \rightarrow (\text{Where } N_{n,m} \text{ was defined in Eq. II.8}) \quad (\text{II.14})$$

It happens in this case that the Φ is not a recursive value, but just a factor depending on (n,m) at each step.

–FOR $m = 0$ (ZONAL COEFFICIENTS)–

$$\Phi_{n,0}^{n+1,1} = \sqrt{\frac{2n+1}{2(2n+3)}(n+2)(n+1)} \quad (\text{II.15})$$

$$\Phi_{n,0}^{n+1,0} = \sqrt{\frac{2n+1}{2n+3}} \quad (\text{II.16})$$

–FOR $m > 0$ (TESSERAL AND SECTORIAL COEFFICIENTS)–

$$\Phi_{n,m}^{n+1,m+1} = \sqrt{\frac{2n+1}{2n+3}(n+m+2)(n+m+1)} \quad (\text{II.17})$$

$$\Phi_{n,m}^{n+1,m-1} = \sqrt{\frac{2n+1}{(2n+3)(n-m+2)(n-m+1)}} \quad \text{if } m > 1 \quad (\text{II.18})$$

$$\Phi_{n,m}^{n+1,m-1} = \sqrt{2 \frac{2n+1}{(2n+3)(n-m+2)(n-m+1)}} \quad \text{if } m = 1 \quad (\text{II.19})$$

$$\Phi_{n,m}^{n+1,m} = \sqrt{\frac{2n+1}{2n+3} \frac{n+m+1}{n-m+1}} \quad (\text{II.20})$$

Examining the equations shown above, it must be noticed, that in the implementation of this algorithm, it must be consider exceptions (due to the normalizing process) at $m = 0$ and at $m = 1$ in the equations.

II.3.1.3 Non-Inertial Acceleration

The terms already computed correspond to the force felt under an absolute reference frame. In this analysis, the fictitious accelerations included here are those corresponding to the rotation of the Earth around its own axis, assuming a constant velocity (no angular acceleration) and no oscillation of this axis.

$$\vec{\omega} = (7.2921150 \pm 0.0000001) \times 10^{-5} \vec{k} \text{ rads/sec}$$

The inertial terms, may be then developed as:

$$\vec{r} = x\vec{i} + y\vec{j} + z\vec{k} \quad (\text{II.21})$$

$$\vec{v} = \frac{d\vec{r}}{dt} = \dot{x}\vec{i} + \dot{y}\vec{j} + \dot{z}\vec{k} + \vec{\omega} \times \vec{r} \quad (\text{II.22})$$

$$\vec{a} = \frac{d\vec{v}}{dt} = \ddot{x}\vec{i} + \ddot{y}\vec{j} + \ddot{z}\vec{k} + \dot{\vec{\omega}} \times \vec{r} + 2\vec{\omega} \times \vec{v} + \vec{\omega} \times \vec{\omega} \times \vec{r} \quad (\text{II.23})$$

Rearranging the Equations in order to integrate them correctly, it is possible to express:

$$\vec{a}_{gravity} - \left[\dot{\vec{\omega}} \times \vec{r} + 2\vec{\omega} \times \vec{v} + \vec{\omega} \times \vec{\omega} \times \vec{r} \right] = \ddot{x}\vec{i} + \ddot{y}\vec{j} + \ddot{z}\vec{k} \quad (\text{II.24})$$

And last equation with the assumptions expressed at the beginning of this section is left as:

$$\vec{a}_{gravity} - [2\vec{\omega} \times \vec{v} + \vec{\omega} \times \vec{\omega} \times \vec{r}] = \ddot{x}\vec{i} + \ddot{y}\vec{j} + \ddot{z}\vec{k} \quad (\text{II.25})$$

Coriolis And Centrifugal Acceleration

In order to reduce computational time, cross products have been further developed such that the following expressions arise:

CORIOLIS FORCE DEVELOPMENT

$$2 \vec{\omega} \times \vec{v} = -v_y \omega \vec{i} + v_x \omega \vec{j} + 0 \vec{k} \quad (\text{II.26})$$

CENTRIFUGAL FORCE DEVELOPMENT

$$\vec{\omega} \times \vec{\omega} \times \vec{r} = -x \omega^2 \vec{i} - y \omega^2 \vec{j} + 0 \vec{k} \quad (\text{II.27})$$

II.3.2 Algorithm Implementation

The implementation of this algorithm has been performed in two different programming languages, C and MatLab.

The implementation has been structured in very similar ways by both methods. The differences among the structure of the code is mainly due to the paths that information must take to move from one point to another (i.e functions to functions). In this way, it is often found pointers and "pointer to pointer" are a common variable in the C programming language.

A great difference occur in the way that spherical harmonics coefficients are loaded in the algorithm. For example the C code loads information from a plain text file whereas such plain text was once formatted and save as a MatLab variable (".mat" files). The last way is loaded in MatLab what is called a "cell array". This data structure for MatLab allows to have stored matrices of different dimensions from cell index to another, which is a very optimized way to store such values and not waste memory. On the other hand in C all coefficients $\bar{C}_{n,m}$ are stored in one single array (of course they are ordered) and the same for $\bar{S}_{n,m}$

II.3.3 Integrator, Trajectory Computation

In this section one computes the trajectories just integrating an IVP with the second order ODE provided by the accelerations outputs.

One could make a study of the speed of the different integrators that it is possible to choose from in MatLab.

In order to do this, a set of integrators were called in a systematic way with a tolerance permission of $1e-12$ (Absolute and Relative) obtaining the following results:

The time of integration were 500 seconds.

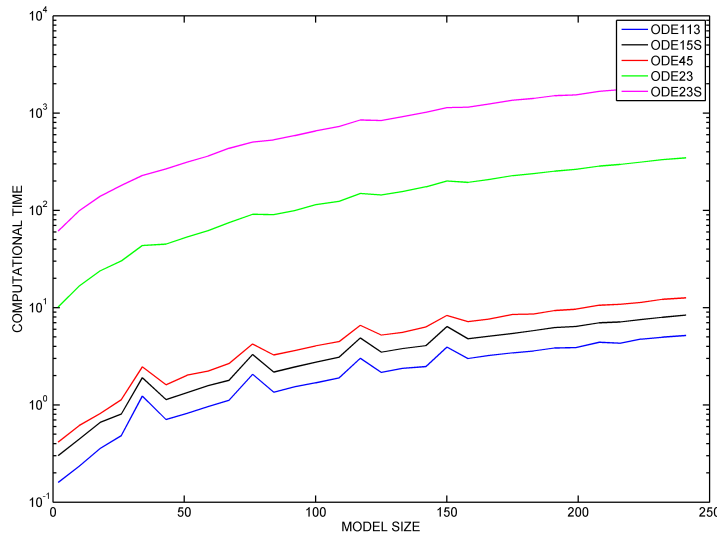


Figure II.3.1: Integrators Efficiency

Last plot shows that the fastest method is ode113, which is an implementation of Adams-Bashforth-Moulton variable order method. This integrator is also one of the recommended for orbit predictions as expressed in ref. [10]

Finally, it must be commented that ode113 is MatLab proprietary function and it is not possible to access to its source code.

The C version of the code lacks of variable step size integrators. A basic

4th order Runge Kutta integrator has been programmed in order to obtain preliminary results.

II.3.3.1 Trajectory Integration Validation in MatLab

It is possible propagate the orbit of a satellite making use of the equations of motion presented previously. For that purpose it is possible to get real information from websites such as "www.celestrak.com" where satellite information is given as a two-line element set. The problem with these sets is that measurements are given with reduced accuracy and with time intervals spaced about a million seconds in time thus it is rather difficult to make a valuation of the accuracy of the results obtained using the previous scheme.

There exists a tool made by NASA called "GMAT" that propagates orbits also taking into account deviations from a perfect spherical body. So it is possible to set up the initial state vector (position and velocity) and launch the integrator. The gravitational body has been chosen to be Earth and the spherical coefficients (mass coefficients) employed in the validation are those used by default in GMAT.

Comparing the results obtained by both means the following figures are used to determine the validity of the method:

First of all, the validity of the results is analysed by means of a normalized difference. A "normalized difference" is simply a relative difference that can be defined as:

$$\text{Normalized Difference} = \frac{|\text{GMAT} - \text{TFG}|}{|\text{GMAT}|} \quad (\text{II.28})$$

It is not possible to call that the error because both results have hypothesis that can be satisfied worse or better but in any case be 100 % true

The previous plot reveals a constant difference for the orbit semi-major axis although there exists a transient oscillatory behaviour at the beginning of the integration period. That may be due to the differences in the integrators used by ode113 (in MatLab) and in GMAT. The comparison among orbital

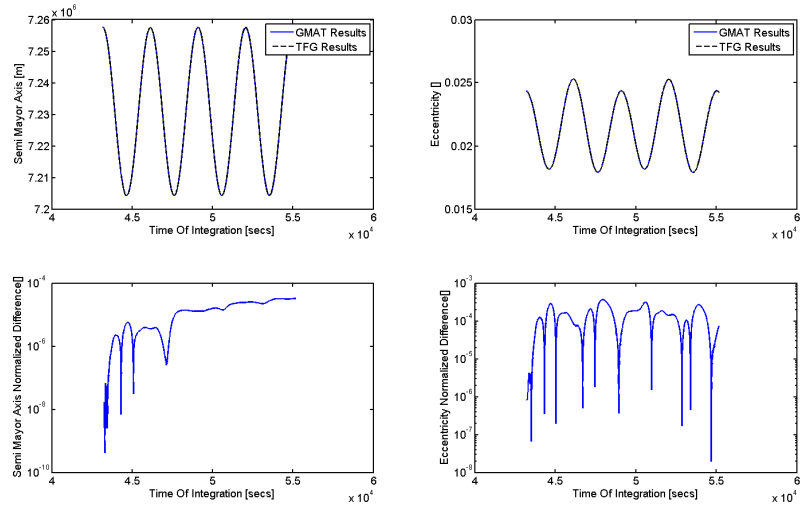


Figure II.3.2: Example of Results Matching in 2 orbital elements

elements has been carried out because historical reasons (Keplerian orbits are usually found in bibliography)

A more intuitive validation metric could be the analysis of the radial distance from the Earth Center Of Mass as seen below:

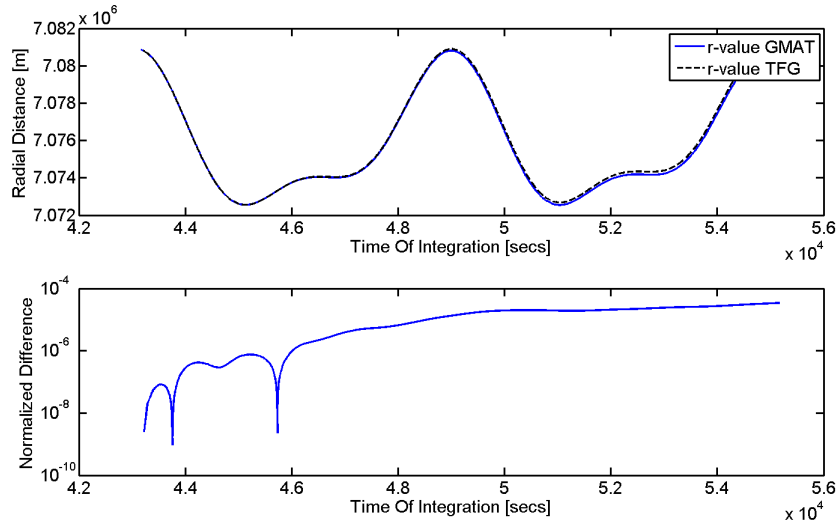


Figure II.3.3: Radial Distance Differences

It is possible to see a transient behaviour of the error being small at the beginning thus reaching a final steady value. The integration time cannot be increased in GMAT thus being not possible to precisely know whether or not the differences keep constant or not.

Part III

DISCRETE BODIES GRAVITATION

Chapter III.1

Introduction

The following algorithm also departs from Eq. I.3 which is again presented below:

$$U = G \int_{\text{Body}} \frac{\rho(\vec{s}) d^3 \vec{s}}{|\vec{r} - \vec{s}|}$$

Instead of solving in terms of coefficients depending on the mass distribution inside the body of study, this method is developed to study the gravity field of a body of general shape, therefore not presenting the divergence problems that were known to occur in the spherical harmonics when calculating the gravitational field inside the so called: "*Reference Sphere*".

In words of the author R.A Werner, ([11]), this method was developed to evaluate the gravitation of irregular-shaped bodies such as comet nuclei, asteroids and small planetary satellites. However the algorithm does not impose any restriction or makes any hypothesis on the size of the body. This method at the cost of computational time and memory, can solve the potential gravitational function for any body size.

This method is intended to be used over volumes that have been discretized. This process consists of reproducing the body surface making use of polygonal faces. In the following pages, without losing generality this work will concentrate in solving the case for a tetrahedron which leads to have free triangular faces (free faces compose the surface of the body). This will not reduce the generality of the method since any polygon can be simplified to a set of triangles.

Chapter III.2

Discrete Body Shape Determination

Shape determination process of our body of study can be retrieved by many different methods. In the following pages we will focus in general methods for irregular bodies and for Earth.

Earth shape can be determined by numerous procedures. From Geographic Information Systems (GIS), to level curves maps or the most simple one the "*WGS84 Reference Ellipsoid*". The later one is the result of producing an ellipsoidal volume that has best fit into millions of topographic points.

The WGS84 ellipsoid is a 2 axis revolution body with the following properties:

1. $a \approx 6378137$ m of Equatorial Radius
2. $f = 1/298.257223563$
3. $b = a(1 - f) \approx 6356752$ m of Polar Radius

With the previous parameters it is easy to build a continuous body surface. The discretizing process has to be done carefully in order not to reach triangles with angles which are too small and which can be the cause of numerical instabilities.

For that reason, Delaunay Triangulation process is applied in order to achieve maximum angles in the triangular faces. In this work MatLab proprietary function "delaunayTriangulation" is used to carry out this process.

The techniques that are presented below are mainly related to the determination of the shape of asteroids and other non regular bodies. This does not mean that they are not valid to determine the shape of Earth, but rather that they are not efficient methods for such purpose.

The shape determination can be carried out via many different techniques. Here are summarized a few of them (source: [3])

- RADAR Imaging
- Light Curves
- Visible/near-infrared (IR) spectroscopy
- Ground-Based thermal infrared
- Space Infrared Telescopes: As Spitzer, WISE, Herschel
- Spacecraft Flyby and Rendezvous missions

All the techniques exposed above are scoped to different target bodies. For example Near Earth Asteroids happen to be too small for optical observations, therefore RADAR Imaging may be much more easy to apply than Light Curves. In fact RADAR images are only possible for bodies located near the radar pulse source as the intensity of the wave decreases to the fourth power with the distance.

For the development of the method of the gravitation of a discrete body, the algorithm that is presented below will not depend on the technique employed in the shape determination, however inaccuracies derived from the shape will have the consequent error in the magnitude of the resulting gravitational field (the solution to the potential function of a polyhedral body is exact once the assumption of constant density is satisfied)

Chapter III.3

Exterior Gravitation Algorithm

In order to start our description of the algorithm, the frame of reference to be employed will be a Cartesian coordinate system with origin at the centre of mass.

For simplicity, the vector \vec{r} that refers to the expression $(\vec{r} - \vec{s})$ in Eq.I.1.

$$\vec{q} = \vec{r} - \vec{s} = \vec{i}\Delta x + \vec{j}\Delta y + \vec{k}\Delta z \quad (\text{III.1})$$

This vector is the relative position from the differential massive element to the field point where the potential function is to be evaluated.

The expression of the potential function, assuming constant density, can be defined only in terms of scalar functions as:

$$U = G\sigma \iiint_V \frac{1}{q} dV \quad (q = |\vec{q}|) \quad (\text{III.2})$$

In order to evaluate the potential function of constant density polyhedron, the idea of transforming the volume integral into a surface integral arises. Therefore one may expect to apply the Gauss-Divergence theorem.

The step of this transformation is not direct. In this work and following results in [11] the integrand is expressed as the divergence of a vector field

still unknown. It is easy to show that such vector field must be:

$$\nabla \cdot \left(\frac{\vec{q}}{2q} \right) = \frac{1}{q} \quad (\text{III.3})$$

Therefore the following development can be followed:

$$U = G\sigma \iiint_V \frac{1}{q} dV = G\sigma \iiint_V \nabla \cdot \left(\frac{\vec{q}}{2q} \right) dV = \frac{1}{2}G\sigma \iint_S \vec{n} \cdot \frac{\vec{q}}{q} dS \quad (\text{III.4})$$

The previous expression is the integral that is intended to do over the body. As the shape of the body is determined by the methods described in Chapter 2 of this part or similar techniques, the shape will be given in terms of polygonal faces. Therefore our final objective is to evaluate the previous surface integral over polygonal faces.

III.3.1 Surface Integral Over a Polygonal Surface

Making use of the linearity of the integral operator, the surface integral of a body composed of polygonal faces can be expressed as the sum of every integral over each face as:

$$U = \frac{1}{2}G\sigma \sum_{f \in \text{faces}} \iint_f \vec{n}_f \cdot \frac{\vec{q}}{q} dS \quad \text{Being the face normal: } \vec{n}_f \quad (\text{III.5})$$

In order to solve the integral over each face, it is mandatory to choose a Cartesian coordinate frame that has the \vec{k} direction parallel to the normal to each face, this can be appreciated in figure III.3.1¹. The directions of the other vectors is unimportant. In this fashion the expression can be simplified as $\vec{n}_f \cdot \vec{q} = \Delta z$ and thus being constant all along the region of integration.

¹In the figure III.3.1, the vector \vec{r}_f is the same as the vector \vec{q}_f in this work

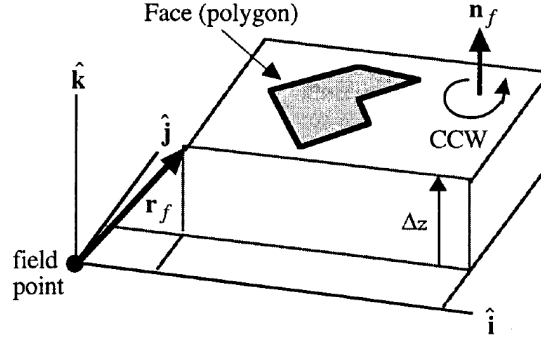


Figure III.3.1: Each face gets its own reference frame.

SOURCE: *Exterior Gravitation of a Polyhedron* R.A Werner [11]

In this way the product of both elements can be taken out. Therefore the expression is left as:

$$U = \frac{1}{2}G\sigma \sum_{f \in faces} (\vec{n}_f \cdot \vec{q}) \iint_f \frac{1}{q} dS \quad (\text{III.6})$$

The mathematical development presented in the previous equations by R.A Werner in [11] are the basis of the mathematical development of the equations presented in the algorithm.

$$\begin{aligned} \iint_S \frac{1}{q} dS &= \iint_S \left(\frac{1}{q} + \frac{\Delta z^2}{q^3} \right) dS - \iint_S \frac{\Delta z^2}{q^3} dS \\ &= \iint_S \left(\frac{q^2 - \Delta x^2}{q^3} + \frac{q^2 - \Delta y^2}{q^3} \right) dS - \Delta z \iint_S \frac{\Delta z}{q^3} dS \\ &= \underbrace{\oint_C \frac{1}{q} (\Delta x d\Delta y - \Delta y d\Delta x)}_{\text{Expression 1}} - \underbrace{\vec{n}_f \cdot \vec{q}_f \cdot \omega_f}_{\text{Expression 2}} \end{aligned} \quad (\text{III.7})$$

Note that Δz is in the reference frame selected $\Delta z = \vec{n}_f \cdot \vec{q}_f$ so that the exact location where \vec{q}_f is pointing to is anywhere in the polygonal face.

The analytical evaluation of each of the members is achieved separately as follows:

Evaluation of Expression 1

By making use of the reference system of III.3.1 (defined per face with \vec{i} and \vec{j} directions defined arbitrarily) it is possible to define an angle between the edge and the face \vec{i} axis α_e . In face plane normal to edge vector will be called \vec{n}_e^f such that:

$$\cos(\alpha_e) = -\vec{n}_e^f \vec{j} \quad (\text{III.8})$$

$$\sin(\alpha_e) = \vec{n}_e^f \vec{i} \quad (\text{III.9})$$

With the previous definitions, it is possible to parametrize the integral along a polygonal face of the mesh as follows:

$$\begin{aligned} \int_{\text{edge}} \frac{1}{q} (\Delta x d\Delta y - \Delta y d\Delta x) &= \int_{\text{edge}} \frac{1}{q} \left[\left(\Delta x_e + s \cos(\alpha_e) \right) \sin(\alpha_e) \right. \\ &\quad \left. - \left(\Delta y_e + s \sin(\alpha_e) \right) \cos(\alpha_e) \right] \quad (\text{III.10}) \end{aligned}$$

The previous expression leads finally to:

$$\int_{\text{edge}} \frac{1}{q} (\Delta x d\Delta y - \Delta y d\Delta x) = \vec{n}_e^f \vec{q}_e^f \int_{\text{edge}} \frac{1}{q} ds \quad (\text{III.11})$$

Where \vec{q}_e^f is a vector from the field point to any point along the edge. Now it can be evaluated the integral of $1/q$ in the previous expression as:

$$L_e^f = \int \frac{1}{q} ds = \ln \frac{a + b + e}{a + b - e} \quad (\text{III.12})$$

Where a,b and e are the distances from the field point to the edge start point, the edge end point and the edge length.

This last equation is singular when $a + b - e = 0$ which will occur at the polyhedron surface. In [11] the author expresses his knowledge about this fact and although no proof is given in principle is expected a well behaviour of the equations. In fact so far both in this work and in the authors work no indetermination nor numerical instability has been appreciated.

Recalling all previous results the final expression of the polyhedron potential is left as:

$$\iint_{\text{Polygon}} \frac{1}{q} dS = \sum_{e \in \text{polygon's edges}} \vec{n}_e^f \cdot \vec{q}_e^f \cdot L_e - \vec{n}_f \cdot \vec{q}_f \omega_f \quad (\text{III.13})$$

Evaluation of Expression 2

Recalling expression 2 in eq. III.7 : $\vec{n}_f \vec{q}_f \omega_f$ it is pursued a geometrical meaning of such expression.

The term ω_f corresponds to the signed solid angle subtended by a planar region when viewed from a field point. Its evaluation over a polygonal face corresponds to solve the following integral:

$$\omega_f = \iint_S \frac{\Delta z}{q^3} dS \quad (\text{III.14})$$

The evaluation of such integral is of very mathematical nature not improving the understanding of the method being described. The mathematical development can be found in [11] and the final results is simply:

$$\omega_f = 2 \arctan \frac{\vec{q}_1 \cdot (\vec{q}_2 \times \vec{q}_3)}{q_1 q_2 q_3 + q_1 (\vec{q}_2 \cdot \vec{q}_3) + q_2 (\vec{q}_1 \cdot \vec{q}_3) + q_3 (\vec{q}_1 \cdot \vec{q}_2)} \quad (\text{III.15})$$

Being $\vec{q}_1, \vec{q}_2, \vec{q}_3$ the vectors pointing from the field point to each of the vertex of the polygonal face. Note that here the problem has been restricted to the case of triangular faces. Again as said previously, this does not restrict

the domain of application of the solution as all polygons can be composed of triangles.

III.3.2 Common Edges, Algorithm Optimization

The equations presented so far have to be evaluated for each of the polyhedron faces. However as the surface is closed, the algorithm would have to go twice through each of the edges. This is seen in the following development:

$$U = \frac{1}{2}G\sigma \sum_{f \in \text{faces}} \vec{n}_f \cdot \vec{q}_f \iint_f \frac{1}{q} dS \quad (\text{III.16})$$

And the integral per polygon has been already solved in terms of sum of terms leaving the expression of the potential as:

$$U = \frac{1}{2}G\sigma \sum_{f \in \text{faces}} \vec{n}_f \cdot \vec{q}_f \left(\sum_{e \in \text{polygon's edges}} \vec{n}_e^f \cdot \vec{q}_e^f \cdot L_e - \vec{n}_f \cdot \vec{q}_f \cdot \omega_f \right) \quad (\text{III.17})$$

In the previous equation it is possible to observe that the sum of terms as: $\vec{q}_{12} \cdot (\vec{n}_A \vec{n}_{12}^A + \vec{n}_B \vec{n}_{21}^B) \cdot \vec{q}_{12}$ will appear repeatedly.

The terms in brackets will be defined as Dyad $E_{12} \equiv \vec{n}_A \vec{n}_{12}^A + \vec{n}_B \vec{n}_{21}^B$. The product of $\vec{n}_f \cdot \vec{n}_f^T$ will be also retained as Dyad defined with F_f . Upon this result the algorithm gets free of nested sums thus keeping the computational time growing linearly with the number of polygons, instead of quadratically as it occurs with nested sums.

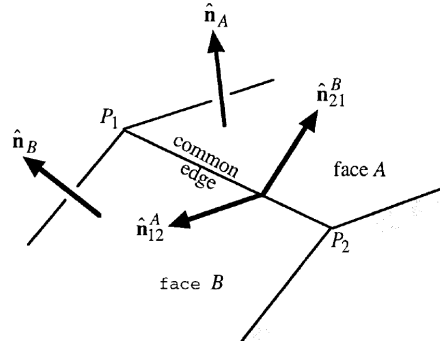


Figure III.3.2: Normal Vectors Diagram.

SOURCE: *Exterior Gravitation of a Polyhedron* R.A Werner [11]

III.3.3 Gradient of Potential Function Acceleration

The computation of the gradient of the potential function although being not straight forward mathematical differentiation, It does not improve the understanding of the algorithm. The mathematical derivation can be found in [11]

The final result for the acceleration vector is then:

$$\vec{a} = \nabla U = G\sigma \sum_{e \in edges} E_e \cdot \vec{q}_e L_e + G\sigma \sum_{f \in faces} F_f \cdot \vec{q}_f \omega_f \quad (III.18)$$

With the solution obtained it is possible to conclude the linearity of the method thus making it prone to implement the equations using parallel computation.

III.3.4 Summary of Results

The gravitational potential of a polyhedron surface can be expressed as:

$$U = \frac{1}{2}G\sigma \sum_{e \in edges} \vec{q}_e \cdot E_e \cdot \vec{q}_e L_e - \frac{1}{2}G\sigma \sum_{f \in faces} \vec{q}_f \cdot F_f \cdot \vec{q}_f \omega_f \quad (III.19)$$

And therefore the acceleration as:

$$\vec{a} = \nabla U = G\sigma \sum_{e \in edges} E_e \cdot \vec{q}_e L_e + G\sigma \sum_{f \in faces} F_f \cdot \vec{q}_f \omega_f \quad (III.20)$$

And the laplacian of the potential function can be expressed as:

$$\nabla^2 U = -G\sigma \sum_{f \in faces} \omega_f \quad (III.21)$$

Note that Laplacian evaluation (Eq. III.21) allows to know whether or not the field point is inside the polyhedron or not. If it vanishes the field point will be outside the the polyhedron.

It is remarkable to say that in terms of computational speed, the signed solid angle ω_f is the one taking higher time in the computation. Per face, ω_f takes 21 products, 20 additions, 1 division and an arctangent (which is a non elemental function). The matrix times vector, for example $F_f \cdot \vec{q}_f$ takes only 9 multiplications and 3 additions per face.

Part IV

SPHERICAL HARMONICS OF A CONSTANT DENSITY POLYHEDRON

Chapter IV.1

Spherical Harmonics Mass Coefficients

Previously in Part II the analytical solution to the gravitational perturbed potential function is given in terms of known coefficients \bar{C}_{nm} and \bar{S}_{nm} that were obtained mainly from databases that process information retrieved by satellites.

This method allows to compute the spherical harmonics mass coefficients derived from a constant density polyhedron. This makes possible to apply the method shown in part II and compare the solutions to that of part III. In the following pages it will be reviewed the process guided in [1] to solve analytically and in exact manner the following equation:

$$\begin{aligned} \begin{bmatrix} C_{n,m} \\ S_{n,m} \end{bmatrix} &= \frac{1}{M} (2 - \delta_{0,m}) \times \frac{(n-m)!}{(n+m)!} \\ &\times \iiint_{\text{Body}} \left(\frac{r'}{a} \right)^n P_{n,m}(\sin(\phi')) \begin{bmatrix} \cos(m\lambda') \\ \sin(m\lambda') \end{bmatrix} dm \end{aligned} \quad (\text{IV.1})$$

Although the method described in part III is general and produces accurate results, the computational cost of the evaluation of the algorithm is constant independently of the location in space is intended to be computed. Therefore, being possible to obtain the spherical harmonics mass coefficients from any

celestial body can optimize the computer resources to produce same accuracy and faster evaluation of the gravitational field. The time required to evaluate the spherical harmonics of a polyhedron is not a relevant parameter as this coefficients are computed only once. In [1] it is expressed that for satellite work solution based on the method exposed in part II has been preferred ("Expansion in Spherical Harmonics").

The first task that is addressed before going directly to the core of the solution is to fix the system of reference. The coordinates are consistent with the frame of reference defined in Eq II.1 to Eq II.3. Note that prime coordinates refer to the location where the mass of the body of study is (polyhedron from now on) and that a is the radius of the so called "Reference Sphere" as defined in Ch. II.1 in part II. The differential element in the previous equation can be expanded in cartesian coordinates as: $dm = \sigma(x', y', z') dx' dy' dz'$ (Still no constant density hypothesis is applied).

From this point on, this work will implement the solution located in [1]. The solution given to these coefficients in such publication scales quadratically with the degree and order due to the presence of nested sums but increases linearly with the number of polygons.

In computational terms it is a highly parellelizable algorithm as the solution to one polyhedron face is independent of the rest of faces and the final result is simply the addition of all of them. Other authors have achieved linear recursive solutions as in [5]. However the authors of the method themselves warn the readers of this last paper about numerical instability of the method, that still remains to be proven. A few more comments about both methods can be better explained after examination of the method in [1].

Within this method, the author R.A Werner defines a new parameter that corresponds mathematically to the integrand in Eq. IV.1 and physically to the contribution of each polygonal face to the final value of the spherical harmonics mass coefficients:

$$\begin{bmatrix} c_{n,m} \\ s_{n,m} \end{bmatrix} = \frac{1}{M} (2 - \delta_{0,m}) \frac{(n-m)!}{(n+m)!} \left(\frac{r'}{a}\right)^n P_{n,m}(\sin(\phi')) \begin{bmatrix} \cos(m\lambda') \\ \sin(m\lambda') \end{bmatrix} \quad (\text{IV.2})$$

Following the recursive scheme that can be applied to Legendre polynomials and the normalizing function shown in Eq. II.8 it is possible to express

recurrence relationships for $\begin{bmatrix} \bar{c}_{n,m} \\ \bar{s}_{n,m} \end{bmatrix}$:

Sectorial Terms, $n=m$ terms

$$\text{For } n = 0 : \begin{bmatrix} \bar{c}_{0,0} \\ \bar{s}_{0,0} \end{bmatrix} = \frac{1}{M} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (\text{IV.3})$$

$$\text{For } n = 1 : \begin{bmatrix} \bar{c}_{1,1} \\ \bar{s}_{1,1} \end{bmatrix} = \frac{1}{\sqrt{3} M a} \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (\text{IV.4})$$

$$\text{For } n > 1 : \begin{bmatrix} \bar{c}_{n,n} \\ \bar{s}_{n,n} \end{bmatrix} = \frac{(2n-1)}{a \sqrt{2n(2n+1)}} \begin{bmatrix} x' & -y' \\ y' & x' \end{bmatrix} \begin{bmatrix} \bar{c}_{n-1,n-1} \\ \bar{s}_{n-1,n-1} \end{bmatrix} \quad (\text{IV.5})$$

Subdiagonal Terms, $m = n - 1$ terms

$$\begin{bmatrix} \bar{c}_{n,n-1} \\ \bar{s}_{n,n-1} \end{bmatrix} = \frac{2n-1}{\sqrt{2n+1}} \frac{z'}{a} \begin{bmatrix} \bar{c}_{n-1,n-1} \\ \bar{s}_{n-1,n-1} \end{bmatrix} \quad (\text{IV.6})$$

Vertical Terms, $n \neq m$ terms

$$\begin{aligned} \begin{bmatrix} \bar{c}_{n,m} \\ \bar{s}_{n,m} \end{bmatrix} &= (2n-1) \sqrt{\frac{(2n-1)}{(2n+1)(n+m)(n-m)}} \frac{z'}{a} \begin{bmatrix} \bar{c}_{n-1,m} \\ \bar{s}_{n-1,m} \end{bmatrix} \\ &\quad - \sqrt{\frac{(2n-3)(n+m-1)(n-m-1)}{(2n+1)(n+m)(n-m)}} \left(\frac{r'}{a}\right)^2 \begin{bmatrix} \bar{c}_{n-2,m} \\ \bar{s}_{n-2,m} \end{bmatrix} \end{aligned} \quad (\text{IV.7})$$

This recursive scheme can be appreciated in the following figure:

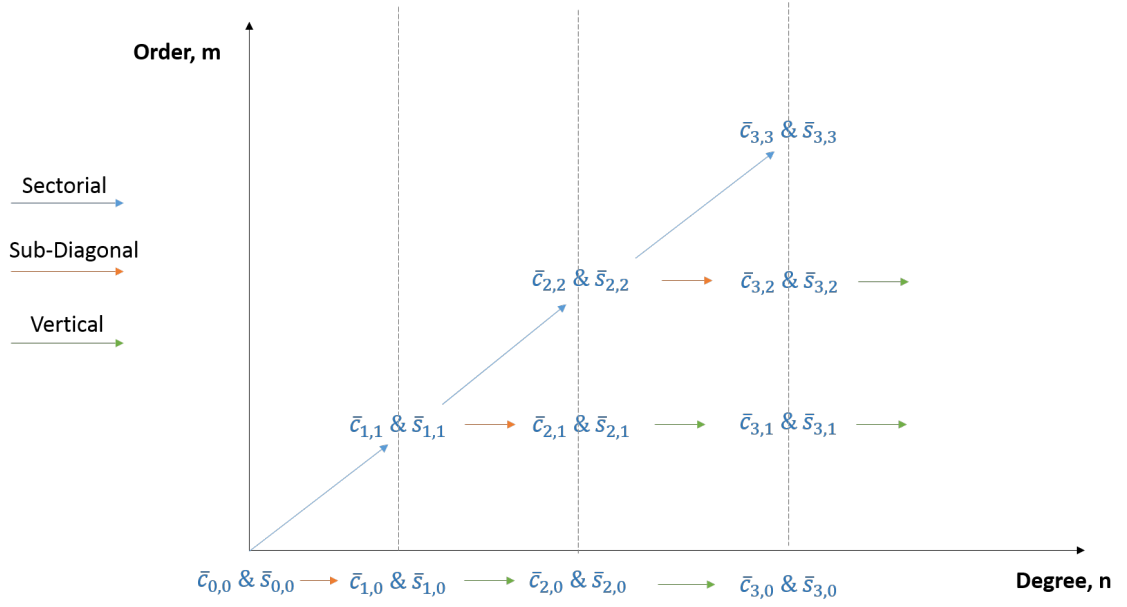


Figure IV.1.1: Recursive Scheme followed when computing the spherical harmonics mass coefficients per polygonal face

Chapter IV.2

Standard Simplex Integration

IV.2.1 Hypothesis

So far it has been possible to derive the recurrent relationships to obtain any order and degree of the coefficients $\begin{bmatrix} \bar{c}_{n,m} \\ \bar{s}_{n,m} \end{bmatrix}$

At this time and aiming to solve the integral in Eq. IV.1, two hypothesis are done:

- The Body Surface is a Polyhedron
Restricted to the case where the faces of a polyhedron is a triangle.
- Constant density

IV.2.2 Change Of Coordinates

Under these assumptions the process to achieved an analytical expression to the integral in Eq. IV.1 is to think that per face (triangle) it is possible to generate a tetrahedron whose 4th vertex is always at the origin of our reference frame.

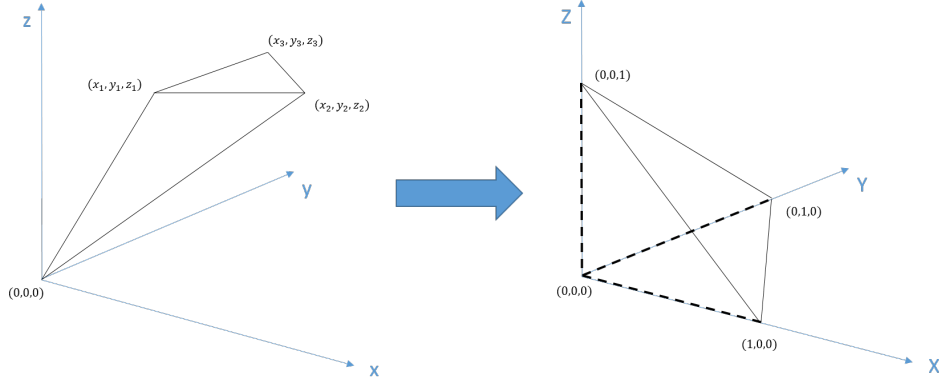


Figure IV.2.1: Transformation of Coordinates to Reach a Standard Simplex

Therefore each face of the polyhedron is divided from the rest in terms of a tetrahedron whose vertices are at: (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) and $(0, 0, 0)$. If keeping the order 1,2 and 3 to rotate counter clockwise around each face of the tetrahedron a determinant can be calculated to make a coordinate transformation to make of this tetrahedron a so called “Standard simplex” with each vertex located at a unit distance over each axis except the 4th vertex which is still at the origin $(0,0,0)$.

The change of variables is the following:

$$x'(X, Y, Z) = x_1X + x_2Y + x_3Z \quad (\text{IV.1})$$

$$y'(X, Y, Z) = y_1X + y_2Y + y_3Z \quad (\text{IV.2})$$

$$z'(X, Y, Z) = z_1X + z_2Y + z_3Z \quad (\text{IV.3})$$

With this change of variables and according to [1] the integrands $\bar{c}_{n,m}$ and $\bar{s}_{n,m}$ are homogeneous polynomials of degree n (as they were previously) but are now defined in terms of X,Y,Z coordinates.

Each of the previous equations is a trinomial (as they have 3 variables). In general terms a trinomial of order and degree n can be expressed as :

$$(aX + bY + cZ)^n = \sum_{i+j+k=n} \alpha_{i,j,k} X^i Y^j Z^k \quad (\text{IV.4})$$

According to the previous expression it is straight forward to obtain the following derivation:

$$\begin{bmatrix} \bar{c}_{n,m}(x', y', z') \\ \bar{s}_{n,m}(x', y', z') \end{bmatrix} \rightarrow \begin{bmatrix} \bar{c}_{n,m}(X, Y, Z) \\ \bar{s}_{n,m}(X, Y, Z) \end{bmatrix} \equiv \sum_{i+j+k=n} \begin{bmatrix} \alpha_{i,j,k} \\ \beta_{i,j,k} \end{bmatrix} X^i Y^j Z^k \quad (\text{IV.5})$$

It is needless to express the necessity of computing the Jacobian determinant after a change of coordinates in an integral. Below it can be seen its expression for each tetrahedron

$$J = \begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{vmatrix} \quad (\text{IV.6})$$

Note that the value of J is constant when integrating along a given standard simplex. One last partial result needs to be obtained. This is obtained in the following section

IV.2.3 Integral Solution

Recalling all previous partial results, the integral in Eq.IV.1 can be developed as:

$$\begin{aligned}
\begin{bmatrix} \bar{C}_{n,m} \\ \bar{S}_{n,m} \end{bmatrix} &= \iiint_{\text{Polyhedron}} \begin{bmatrix} \bar{c}_{n,m} \\ \bar{s}_{n,m} \end{bmatrix} dm \\
&= \sigma \sum_{\text{Tetrahedrons}} \iiint_{\text{Tetrahedron}} \begin{bmatrix} \bar{c}_{n,m}(x', y', z') \\ \bar{s}_{n,m}(x', y', z') \end{bmatrix} dx' dy' dz' \\
&= \sigma \sum_{\text{Tetrahedrons}} \iiint_{\text{Std. Simplex}} \begin{bmatrix} \bar{c}_{n,m}(X, Y, Z) \\ \bar{s}_{n,m}(X, Y, Z) \end{bmatrix} J dX dY dZ \\
&= \sigma \sum_{\text{Tetrahedrons}} \left(J \sum_{i+j+k=n} \begin{bmatrix} \alpha_{i,j,k} \\ \beta_{i,j,k} \end{bmatrix} \right. \\
&\quad \left. \iiint_{\text{Std. Simplex}} X^i Y^j Z^k dX dY dZ \right) \tag{IV.7}
\end{aligned}$$

The triple integral into previous result does have an elegant analytical solution as can be found in [1].

$$\iiint_{\text{Std. Simplex}} X^i Y^j Z^k dX dY dZ = \frac{i! j! k!}{(i+j+k+3)!} = \frac{i! j! k!}{(n+3)!} \tag{IV.8}$$

So that it is possible to express finally the solution to Eq. IV.1 as:

$$\begin{bmatrix} \bar{C}_{n,m} \\ \bar{S}_{n,m} \end{bmatrix} = \sigma \sum_{\text{Tetrahedrons}} \left(\frac{J}{(n+3)!} \sum_{i+j+k=n} i! j! k! \begin{bmatrix} \alpha_{i,j,k} \\ \beta_{i,j,k} \end{bmatrix} \right) \tag{IV.9}$$

The previous equation solves in a exact way the value of the spherical harmonics mass coefficients coming from Eq. IV.1. The solution is exact once the hypothesis in IV.2.1 are satisfied. As can be seen the sum over the

number of tetrahedrons scales linearly the computational time. However the increase in the order and degree scales quadratically in computational time. The same conclusions can be managed from the memory consumption point of view.

IV.2.4 Algorithm Implementation Comments

The MatLab code accompanying this report obtains the solution to Eq. IV.9. In the process the reader could at first sight visualize a 3D matrix (a tensor formally speaking) where coefficients $\alpha_{i,j,k}$ and $\beta_{i,j,k}$ were stored using i,j,k as indices. However one should never forget the constraint $i + j + k = n$ thus being possible a reduction in the size of the matrix allocating the values of such coefficients¹. As it was expressed previously, each trinomial has the same degree as the corresponding spherical harmonic coefficient. Due to the constraint going from a tensor to a matrix, this will be $n \times n$ being filled only a triangular portion of it, thus being sparse.

$$\begin{bmatrix} \alpha_{0,0,k} & \alpha_{1,0,k-1} & \alpha_{2,0,k-2} \cdots & \cdots & \cdots & \alpha_{n,0,0} \\ \alpha_{0,1,k-1} & \alpha_{1,1,k-2} & \alpha_{2,1,k-3} \cdots & \cdots & \alpha_{n-1,1,0} & 0 \\ \alpha_{0,2,k-2} & \alpha_{1,2,k-3} & \alpha_{2,2,k-4} \cdots & \alpha_{n-2,2,0} & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \alpha_{0,n,0} & & & & & \end{bmatrix} \quad (\text{IV.10})$$

And same applies for the coefficients in $\beta_{i,j,k}$

In addition, it has to be said that trinomial multiplication is indeed a two dimensional convolution. In Eqs. IV.3 through IV.7 the variables x', y' and z' are all trinomials. So are $\bar{c}_{n,m}$ and $\bar{s}_{n,m}$. Being our aim is to get the α and β coefficients, a convolution must be performed in order to keep track of them².

¹Such reduction is given by writing the value of the coefficient, for example $\alpha_{i,j,k}$ at simply (i,j) in the 2D matrix

²The algorithm needs to be fed by the polynomial coefficients rather than the product value

Part V

APPLICATIONS

Chapter V.1

Earth Gravity Field And WGS84 Ellipsoid

The Earth gravitational field as well as the validity of using the Ellipsoid WGS84 are analyzed in this chapter.

In part II the aspherical potential function was solved in terms of the spherical harmonics mass coefficients that were known from satellite observation and stored in databases.

In chapter III.2 in part III it was defined the WGS84 ellipsoid widely used nowadays in cartographic services. This ellipsoid can be easily handled making use of parametric equations such as:

$$x = a \cos(\phi) \cos(\lambda) \tag{V.1}$$

$$y = a \cos(\phi) \sin(\lambda) \tag{V.2}$$

$$z = b \sin(\phi) \tag{V.3}$$

Being a and b the equatorial and polar radius. With the previous description of the ellipsoid it is possible to generate a mesh taking into account a Delaunay triangulation to produce an optimal mesh of triangles (angles are maximized).

V.1.1 Gravity Intensity Variation

One direct and simple applications is to obtain the gravity intensity variation from the Earth surface up to some level of, for example “Geopotential-height” in space.

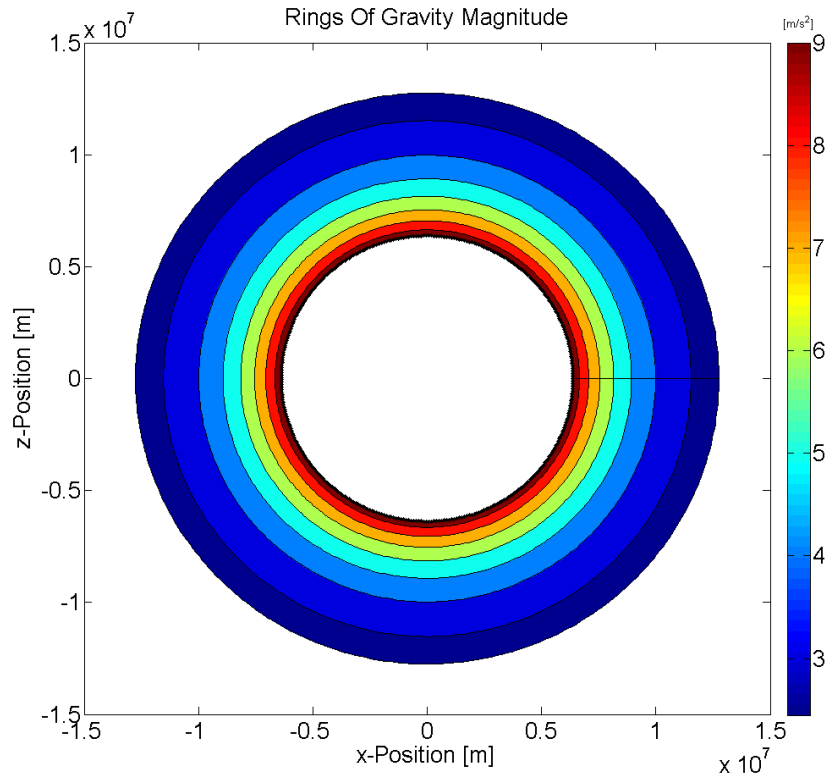


Figure V.1.1: Contours Generated using EGM96 spherical harmonics mass coefficients (degree 360). Gravity intensity is in m/s^2

There is not much to comment from the previous plots as the quadratic reduction in gravitational field intensity a well known fact. However what can be done is to study the relative accuracy of the models previously described and compare the results.

V.1.2 WGS84 Ellipsoid Gravity Field

It is possible to compare the relative difference among the two solutions to the gravitational potential function. Furthermore, it will be possible to compute the spherical harmonics of the discretized body and make a comparison using the 2 sets of spherical harmonics coefficients.

V.1.2.1 The Mass Constraint

The algorithms here presented are highly dependent on the mass and constant density of the bodies. When applying a discretization process to the ellipsoid the volume may change thus 3 things can be done:

- Correct density
- Correct mass
- Correct volume

As we want to keep our density and mass constant in our calculations so that the gravity field is fed by the same parameters, our decision is to correct the volume.

Since an ellipsoid is a convex surface, so is our mesh. Thus being a convex polyhedron the volume can be computed as the sum of the volume of each of the tetrahedrons, not adding the same volume twice. A schematic representation is seen in the left part of figure IV.2.1.

A common way to compute the volume of a tetrahedron with edges that have different dimensions is and with vertices at coordinates (x_i, y_i, z_i) :

$$V_{\text{tetrahedron}} = \frac{1}{3!} \begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix} = \frac{1}{6} \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} \quad (\text{V.4})$$

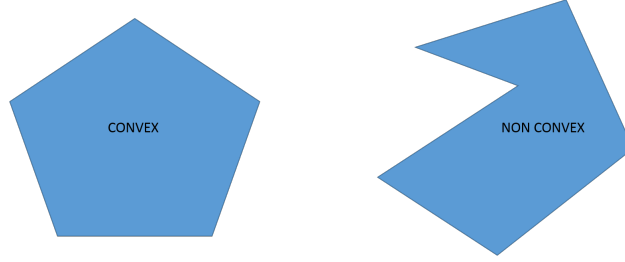


Figure V.1.2: Convex and Non Convex definitions

The simplification done in the previous equation is possible simply because one vertex is always at the origin of the frame of reference therefore due to basic algebraic properties the matrix dimensions inside the determinant can be reduced.

Therefore the final volume of a tetrahedron would be:

$$V_{\text{polyhedron}} = \sum V_{\text{tetrahedron}} \quad (\text{V.5})$$

Once the volume of the discretized surface is found can be compared to the original WGS84 Ellipsoid using the well known formula for a two axis ellipsoid applies as: $V_{\text{WGS84}} = 4/3\pi a^2 b$. Now a scale factor can be applied to the location of each vertex as:

$$\text{Scale Factor} \rightarrow SF = \left(\frac{V_{\text{WGS84}}}{V_{\text{discrete}}} \right)^{\frac{1}{3}} \quad (\text{V.6})$$

And now each pair of coordinates is replaced as:

$$(x_i, y_i, z_i)_{\text{Corrected}} = SF \cdot (x_i, y_i, z_i) \quad (\text{V.7})$$

Density, Mass and Reference Sphere

It rests to define the numerical value of the mass used in the computations of the gravity field produced by WGS84 ellipsoid.

- Earth Mass = $M_{\oplus} = 5.97219 \times 10^{24}\text{Kg}$
- The density is computed dividing the mass by the WGS84 ellipsoid volume (as computed previously):

$$\sigma = 5.51343 \times 10^3 \frac{\text{Kg}}{\text{m}^3}$$

- The reference sphere is computed as the greatest distance from the origin of reference frame to each of the polyhedron vertex. This value may change from polyhedron to polyhedron (variations in the number of polygons, P).

V.1.2.2 Spherical Harmonics Mass Coefficients of Discrete Body

Using the method described in part IV it is possible to compute the spherical harmonics of discretized body. A few analytical solutions must be in the mind of the reader, these are found in [6].

$$\begin{bmatrix} \bar{C}_{0,0} \\ \bar{S}_{0,0} \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow \text{Zero degree coefficients} \quad (\text{V.8})$$

$$\begin{bmatrix} \bar{C}_{1,m} \\ \bar{S}_{1,m} \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow \text{First degree coefficients} \quad (\text{V.9})$$

$$\bar{S}_{n,0} \equiv 0 \rightarrow \text{First degree coefficients} \quad (\text{V.10})$$

With the previous analytical solutions, in this work we conclude the following:

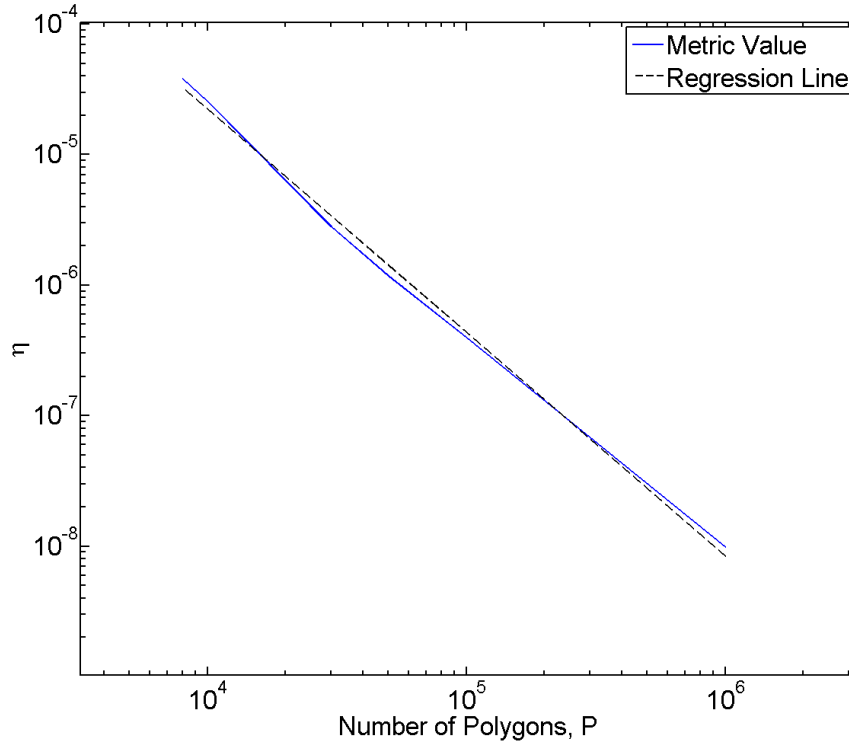
- Solution in V.8 must be always satisfied even in a discretized body. The algorithm exposed in part IV. If not a perfect matching to 1 is obtained can be due to discrepancies between mass, volume, density in the discretized body.
- Solution in V.10 it is also a must even in a discretized body. Tracking the algorithm, this value comes from bottom equation in IV.3.

- Solution in V.9 allows few more arguments. The truth about these set of terms is that in the end they depend among sum and rest all along the summation of terms in the recursions. In [1] it is possible to observe the solutions obtained by the author of the algorithm where they are far from being zero. In fact in [1] when performing an analysis of an unknown body the coefficient $\bar{C}_{1,0} = 0.285162$ which is far from a so called "numerical zero" (which could be defined as 10^{-16} or smaller). However the fact that we are using the WGS84 it is possible to vary the mesh resolution, making it finer, it is expected a convergence to zero in those terms.

Defining as a metric the parameter

$$\eta = \sqrt{\sum_{m=0}^{m=1} (\bar{C}_{1,m})^2 + \sum_{m=0}^{m=1} (\bar{S}_{1,m})^2}$$

It is possible to observe the convergence of values as:



The regression line (logarithmic regression line) shows an almost quadratic convergence (for this WGS84 ellipsoid):

If P is the number of polygons:

$$\eta \approx 155.4652 \times P^{-1.7710} \quad (\text{V.11})$$

It can be said even more about these set of coefficients (those in eq.V.9). In [6] it is possible to find that the magnitude of this coefficients is directly proportional to the relative distance of the origin of the frame of reference chosen, to the centroid of the body. Thus having done the spherical harmonics mass coefficients evaluation over the original WGS84 parametrized surface these results would have been set automatically to zero. When discretizing the surface, a process called "Delaunay Triangulation" was applied so that triangular faces have an optimal shape reducing numerical instabilities. At this point, it is not possible to know how uniform are spread the triangular faces along the surface, therefore shifting the centre of mass. The previous description can be seen graphically in the following figure:

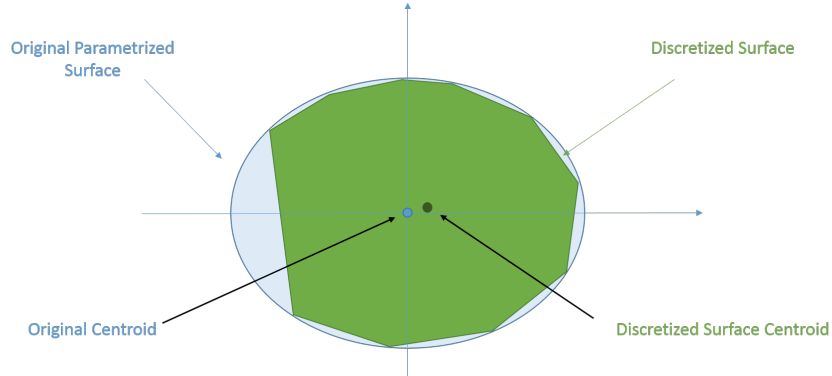


Figure V.1.3: Discretization affecting geometrical properties of the bodies

V.1.2.3 Evaluation of Gravity Field

Recalling the purpose that was being pursued, it will be compared in the next figure the results from 2 methods, using spherical harmonics (Part II) and using the discrete body method (Part III). In the case of spherical harmonics we will use the spherical harmonics mass coefficients computed by NASA in the Geoid Model in 1996 (model called EGM96 up to degree 360) and also those computed by us using our discrete body.

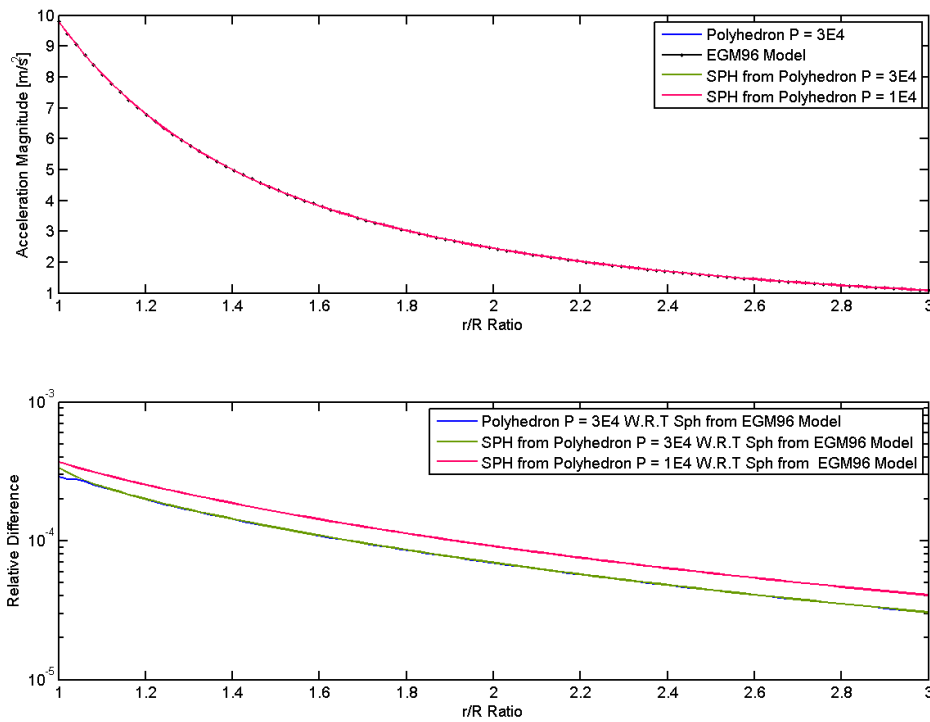


Figure V.1.4: Comparison among the solutions proposed in this report.
Only up to degree 20 has been used

The polyhedron used in the previous figure had a number of polygons of 10^4 . Although it is possible to compute spherical harmonics mass coefficients for, in principle, any mesh size (due to fact that the algorithm is a simple sum, you can compute for a few faces add up, then compute for another few more and so on ¹).

¹This is best explained in the Software Manual in Annex A

However the evaluation of the polyhedron algorithm needs to generate a check matrix to know about edge-sharing faces (Dyads Computation). Even though a sparse treatment can be given to this matrix it has been impossible to compute for a size greater than $P > 3 \times 10^4$ (in normal computers).

Furthermore it is a nice result to see a perfect match of the results between the two methods for a common mesh (lines red and blue in the bottom part of Figure V.1.4). Also it is remarkable to see how differences are smaller for bigger mesh sizes (comparing with EGM96 model which is computed by NASA)

Lastly, the previous figure has shown increasing differences when the Gravitational Potential is evaluated closer to Earth surface. A bigger difference means no bigger error since inside the reference sphere, the spherical harmonics must diverge. This suggests making a divergence analysis at the Earth Surface.

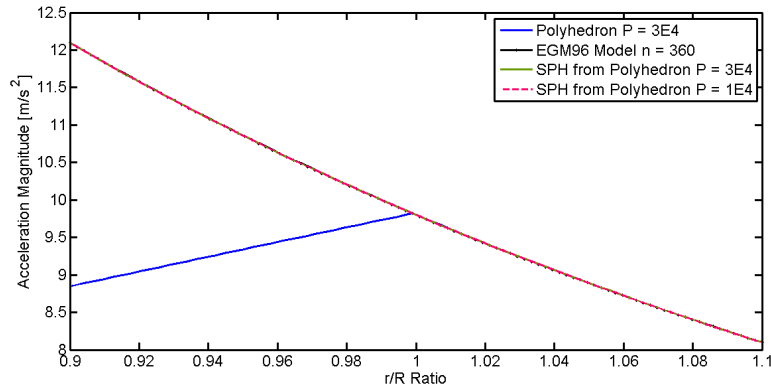


Figure V.1.5: Divergence at the Earth Surface.
Only up to degree 20 in spherical harmonics has been used

From the previous figure, one must first of all ~~must~~ distinguish two regions:

$$\frac{r}{R} > 1 \quad \rightarrow \text{Validity of methods in Parts II and III} \quad (\text{V.12})$$

$$\frac{r}{R} \leq 1 \quad \rightarrow \text{Validity of method in Part III} \quad (\text{V.13})$$

Even not pursuing the evaluation of the gravity field inside the ellipsoid, it

is remarkable to express the conditions of validity as the usage of spherical harmonics solution near body surface may lead to not reliable data.

In [2] the author makes an analysis of the optimum models used to determine a fast and reliable evaluation of the gravity field around an asteroid. In that work, it is recommended to use spherical harmonics in the region outside the reference sphere and use the polyhedral method inside. However in order to produce a faster evaluation of the gravity field expresses the need to pre-calculate a set of points in space and then perform interpolations.

After these comments on the methods, in this work one realises that the speed of the divergence is directly related to the degree of the spherical harmonics model. Furthermore, the higher the degree, the faster the divergence. Increasing the degree used to compute EGM96 up to 360 (maximum possible) one see a much faster divergence as below:

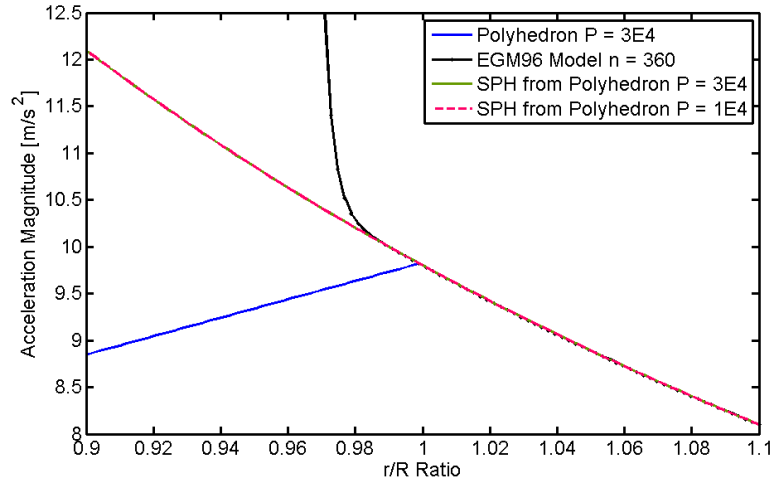


Figure V.1.6: High degree divergence (Polyhedron Derived Spherical Harmonics Mass Coefficients are of degree 20)

Previous figure shows that the divergence always occur at $r/R < 1$. In the next figure, a comparison among spherical harmonics derived from finer and finer meshes is shown:

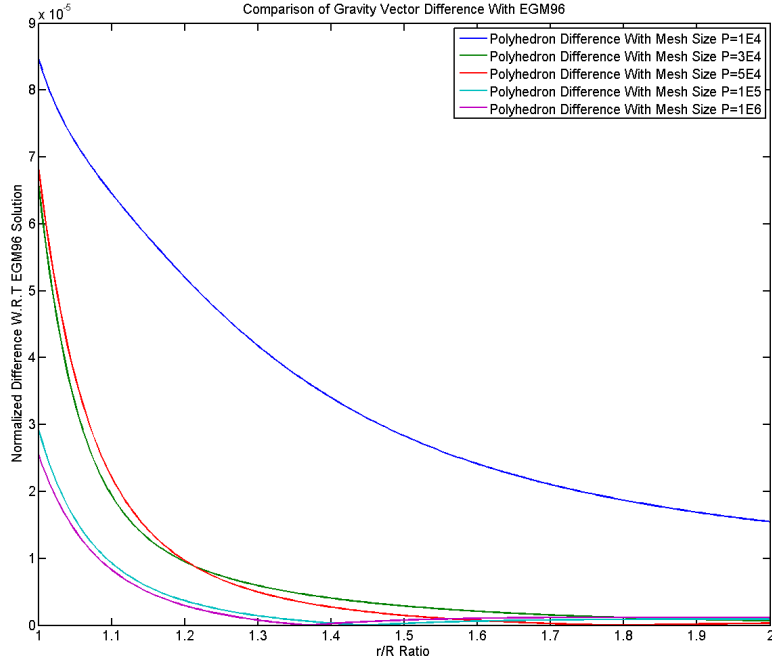


Figure V.1.7: Comparison Of Mesh Size Convergence To EGM96 Values.
Only up to degree 20 in spherical harmonics has been used

The divergence occurs for not being valid the equation II.5 since condition in II.2.1 ($|r| > |s|$) is not satisfied.

After all this plots, It is possible to conclude that the mesh size is a much more important parameter that the degree of the model used in spherical harmonics.

Chapter V.2

Asteroid Castalia Analysis

This analysis is performed to evaluate the gravitational field around asteroid “4769 Castalia”. Also this asteroid serves a test body to check the divergence of the spherical harmonics coefficients in non-convex geometry.

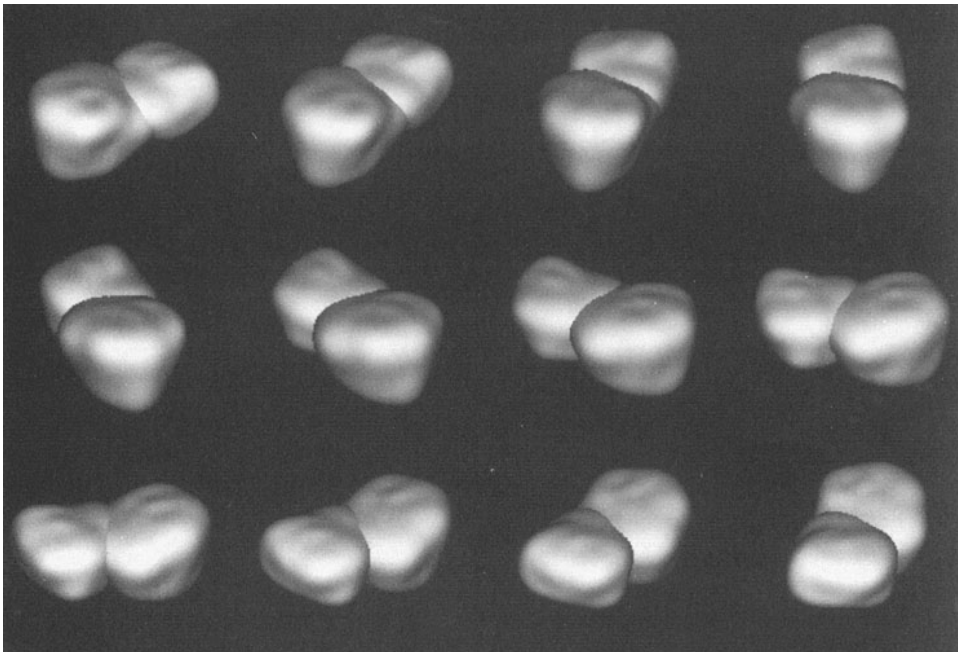


Figure V.2.1: Series of delay-doppler radar images of the asteroid 4769 Castalia

SOURCE: history.nasa.gov

V.2.1 Gravity Field Intensity Near Surface

By using the method exposed in Part III and using the data from this asteroid found in [11] it is possible to define the required body parameters to evaluate Castalia Asteroid gravity field:

- Mass = $M_{\oplus} = 5 \times 10^{11} \text{Kg}$
- The value of density is found in [11]: $\sigma = 2.1 \times 10^3 \frac{\text{Kg}}{\text{m}^3}$. Note that as this is a non-convex polyhedron it is not possible to use the method shown in Eq. V.4
- The reference sphere is computed as the greatest distance from the origin of reference frame to each of the polyhedron vertex. As the mesh of the asteroid is unique (at least it has only been found 1), the reference sphere is:

$$a = 881.11 \text{ m}$$

The polyhedral description of the asteroid can be found on internet¹. The usual absolute reference frame takes the z-direction as the direction of the angular velocity vector which is nearly constant in direction.

In figure V.2.2 the gravity magnitude contours² are found over a cross section of the equator. It is remarkable to say how constant magnitude lines seem to follow the surface concavities. Also important to say how weak is the gravity on the surface of the asteroid. The previous results are the absolute acceleration.

Due to this fact as well as the usual high speed spin that asteroids bodies have, the non inertial forces can have unusual level of importance. In NASA web site it is possible to find the sidereal time period of the asteroid, which is 4.07 h thus rotational speed is:

$$\omega \approx \frac{2 \pi}{4.07 \times 3600} = 4.288 \times 10^{-4} \text{ rads/s} \quad (\text{V.1})$$

¹The source file with the information used in this work is attached to this document

²In the figure, the region in black corresponds to the asteroid cross section.

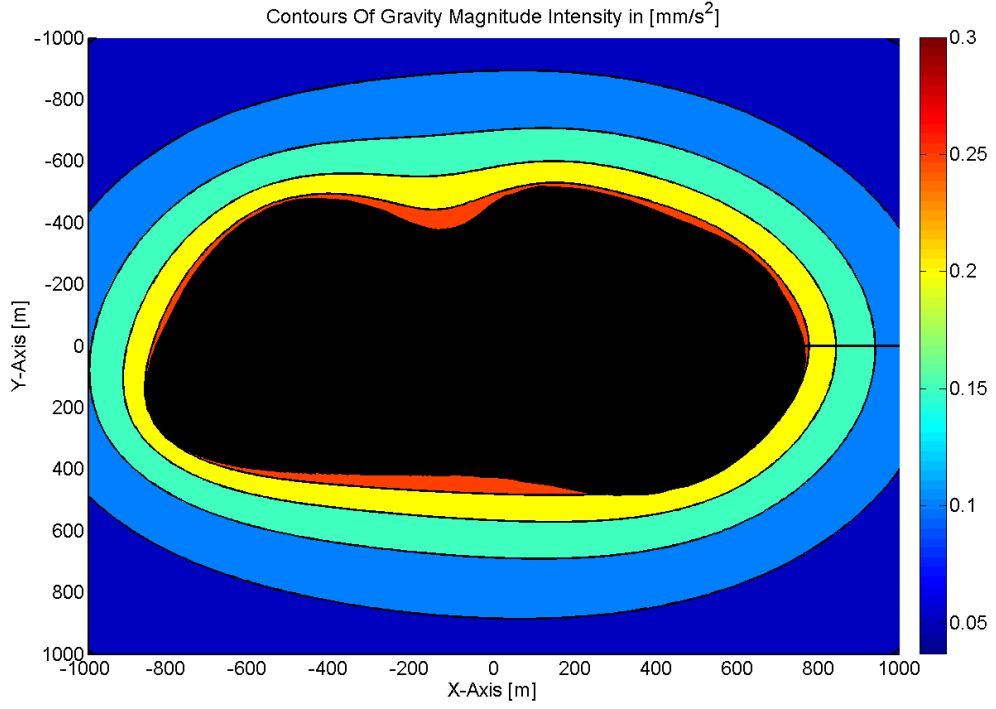


Figure V.2.2: Asteroid 4769 Castalia Gravity Contours in [mm/s²]. X-Y cross section

And the magnitude of the centrifugal acceleration:

$$\omega^2 a = 1.620 \times 10^{-1} \text{mm/s}^2 \quad (\text{V.2})$$

This about half of maximum gravitational intensity that is found on the surface at the equator.

Besides, in this section it will be compared the results found in this work with those of in [11]. In figure V.2.3 it is possible to see the original solution.

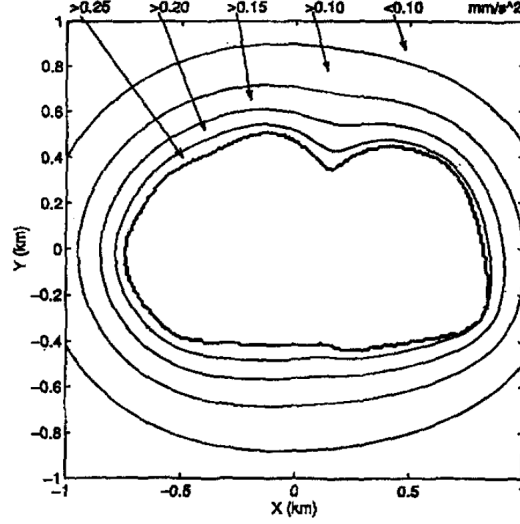


Figure V.2.3: Original Solution found in [11]

Together with that figure, it is possible to measure the (x,y) coordinates of the lines of constant magnitude. In this work this was done with the line at $g = 0.25 \text{ mm/s}^2$. As it does not have to be the case in which the values measured have been computed, a linear 2D interpolation was done to calculate the relative difference to that value at the measured coordinates. In figure V.2.4 the comparison of the value of gravity in this work to that of found in [11] is seen.

Finally, the comparison is not among the exact polyhedral description of the Asteroid. Werner in his paper talks about a polyhedron composed of $P = 3300$ polygons, whereas the one used in this work has 4092. In the previous chapter it was seen that the number of polygons was the main parameter affecting results, thus this percentage error seems to be acceptable.

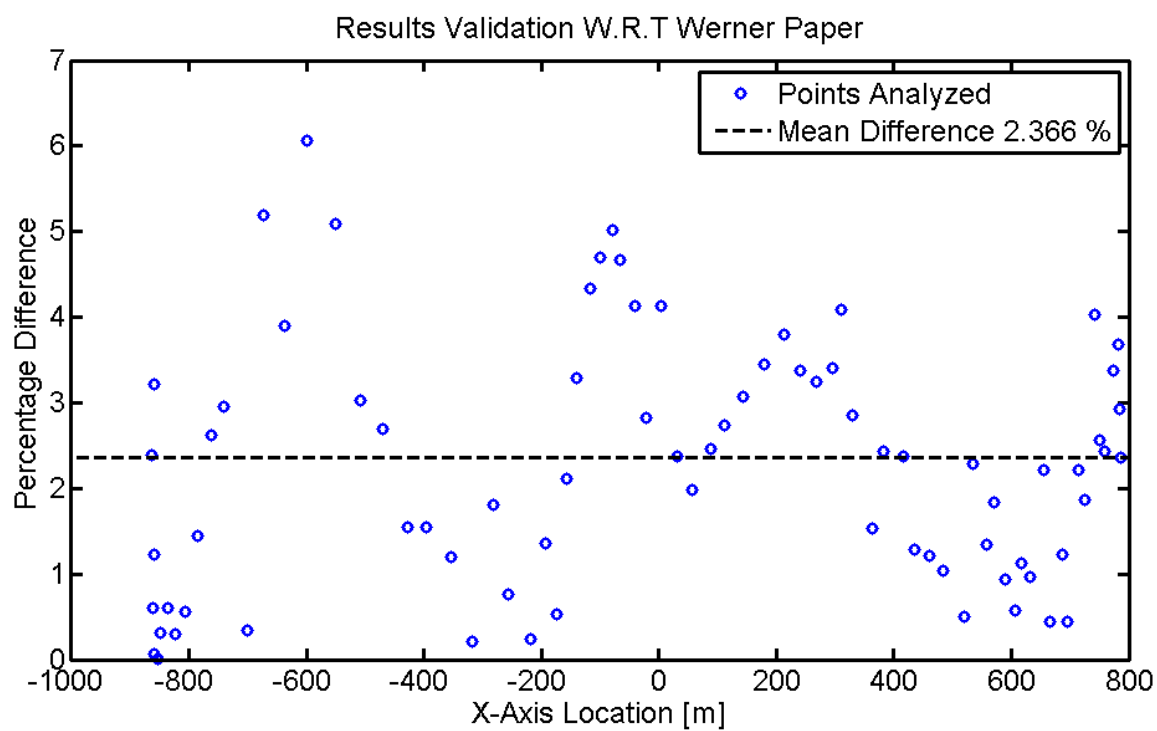


Figure V.2.4: Validation and Comparison of Results

V.2.2 Methods Comparison

In the following a comparison among the methods in parts II and III will be done. Most of the attention will be paid to the regions where both methods have radically different solutions.

The spherical harmonics mass coefficients have been derived from the method exposed in part IV using the asteroid data that can be found in this chapter. Performing the comparison leads to the results shown in figure V.2.5.

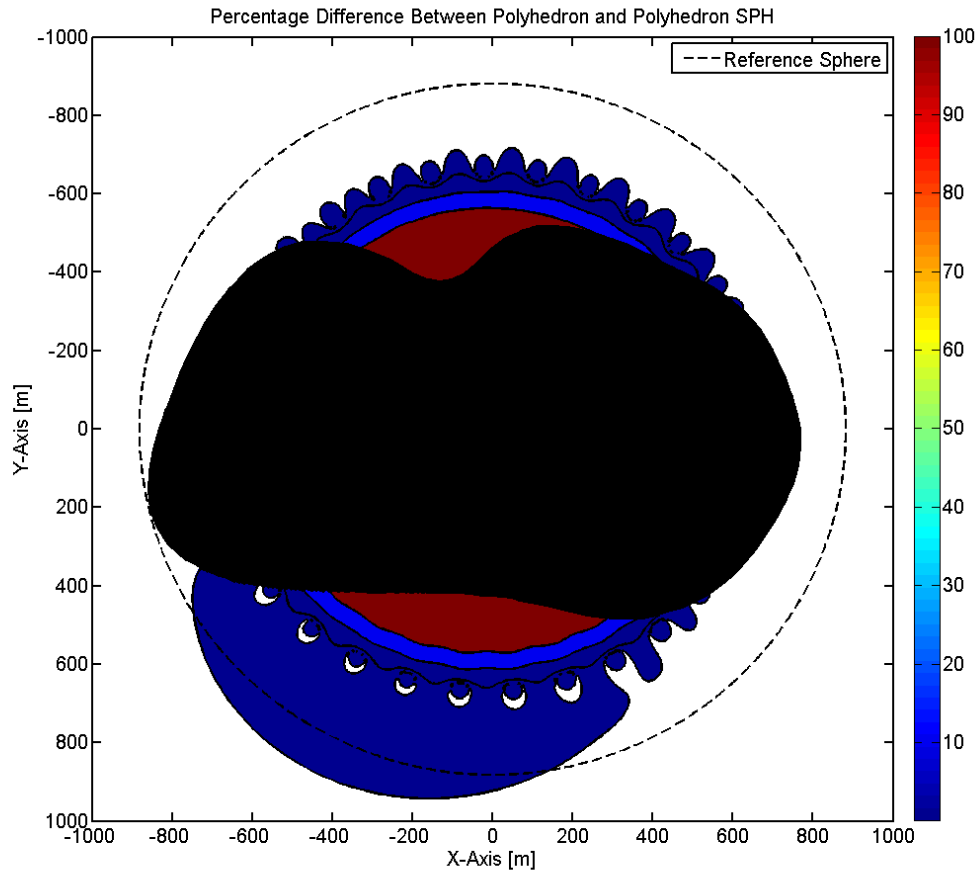


Figure V.2.5: Percentage Difference between methods in parts II and III

First of all, it must be comment that the region in white correspond to differences below 0.1 %. Thus errors bigger than such value are those coloured.

Furthermore in the figure, it is possible to observe divergence of the spherical harmonics solution inside the reference sphere producing bulges. This description of the difference among the methods and its divergence can also be found in [11]. Furthermore the regions in red, those closer to the asteroid surface at concave regions are those producing faster divergences. It is also remarkable to note that those segments of the cross section closer to the reference sphere experience almost no divergence.

Part VI

CONCLUSIONS

CONCLUSIONS

There are numerous conclusions that derive from a wide study as this was. This must not be understood in any case as results obtained from using numerical analysis to solve equations but rather as numerical evaluation of analytical solutions.

Solutions based on spherical harmonics as presented in part II leads to a simple normalized recursive scheme that can be directly implemented without further considerations. This however seems to impose a big restriction in the tendencies of improvement of today's computers, **parallelism**.

The solution presented in part III is highly parallelizable. First of all, looking at equations III.19 and III.20 both are described in terms of sums. Again it is noted the cost sink that the term ω_f is. These expressions are the perfect case scenario for GPU computing (in clear agreement with [2]). The evaluation of all ω_f terms takes few computer resources but big amount of time due to serial computation. Taking advantage of this massively parallelization, the time needed for the evaluation of this term can be minimised in time by hardware. In the present job taking advantage of MatLab ability to manipulate matrix multiplications:

$$\sum_{e \in \text{edges}} E_e \cdot \vec{q}_e = \sum \left(\begin{bmatrix} E_1 & \cdots & E_n \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix} \right) \quad (\text{VI.1})$$

Same applies for the faces product.

Note that the number of edges in a polyhedron can be computed using Euler formula as:

$$n_{\text{edges}} = P + V - 2 \rightarrow \text{Being } P \text{ the number of faces} \\ \text{And } V \text{ the number of vertices} \quad (\text{VI.2})$$

Large matrix multiplications can also be splitted and parallelized thus making this algorithm not only for precise near the surface of the body of study, but

also very probably of faster when implemented in parallel. The so called futurework of these solutions should be oriented in this direction.

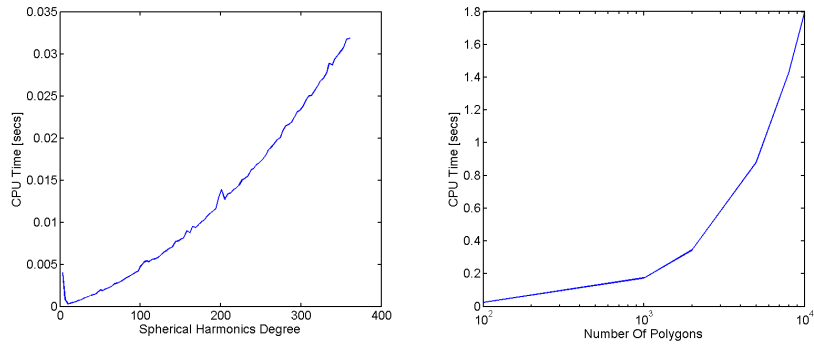
Apart from hardware or software discussions. It of high importance to note the solutions to this work are all based on the assumption of constant density. Density measurements requires fly-by missions as expressed in [11] which compute the spherical harmonics mass coefficients to later reverse this information and retrieve back density distribution data.

Furthermore, it is a fact that the polyhedral algorithm is linear in mass, thus being possible to simulate non-constant density polyhedrons by making use of rings or closed bodies of constant density, as if there were a body inside another one.

Besides, It is for me an elegant and incredible result the solution appearing in Eq. IV.8 thus being possible to express the solution to a triple integral over a standard simplex as a simple function evaluation. This result makes possible to have an analytical expression for the spherical harmonics mass coefficients derived from a polyhedron. Not being less importance the change of coordinates to a standard simplex.

The solutions presented in this job helps to model the gravitational field of an irregular body therefore making possible to integrate (this time numerically) the equations of motion and propagate accurately orbits. Again, effect different from gravitational nature have not been considered (atmospheric drag or solar radiation pressure).

After having compared all methods but not having said anything about timing, below it is found time comparisons under same hardware and with single-threaded algorithms.



The previous times were computed using the matlab implementation.

Conclusions after all this work indicates that gravity field outside the reference sphere should be computed using spherical harmonics because of the high accuracy and low computational time required. However when reaching the surface of the body, polyhedron gravitation should be used instead, increasing accuracy and stability of the method, at a cost of CPU time.

For the future, simulation of launch sequences and landings on bodies together with the implementation of control systems and thrust physics may allow to perform full detail simulation of a space mission. This work has achieved the very first step in recreating precisely the gravity field. Furthermore, another student of this thesis director is using the gravitational model here developed to feed the gravity vector in a spaceship launch. I want to thank to him in helping to find bugs and test the implementation used here.

With these last words, I would like to thank Manuel Sanjurjo Rivo and Daniel González Arribas for their effort, patience, constructive criticism which has been a constant over the past year since I started working with them.

Part A

ANNEX, Software Manual

Chapter A.1

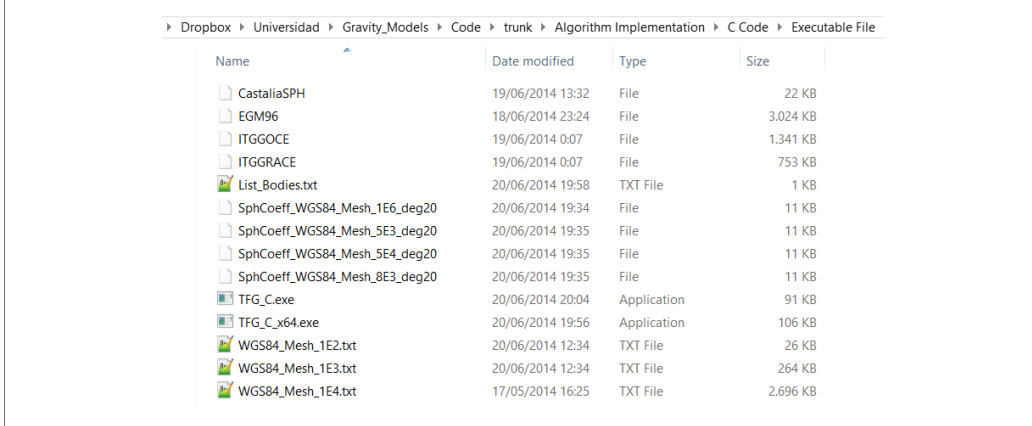
Implementation in C

The algorithms exposed in Parts II, III have been implemented in C. There exists a compressed file attached to this document called “Algorithms Implementation”. Accessing to the path “\Algorithm Implementation\C Code” three different folders arise:

- “c_Code Source Files”: In this folder it is possible to find the source code files that can be compiled to create and execute the algorithm operations.
- “Executable File”: In this folder it is possible to find a compilation of the previous source files. 2 Different executable files are found. One compiled in 32 bits (TFG_C.exe) and another in 64 bits (TFG_C_x64.exe). I would highly recommend to use the 64 bits version of the code if possible. These algorithms require a big amount of memory. Since the limit of 32 bit programs is 2 GB, it is possible to block the program operation when the algorithm tries to allocate more than this memory limit. The rest of the files found in this folder are described later in this chapter.
- “Visual Studio Project”: In this folder it is stored the original visual studio project that was the tool employed to develop the source code and compile it. If changes in the source code are going to be done, I would highly recommend to open this project with visual studio and make the changes inside of it since the code is organised in folders and has some hierarchy which cannot be appreciated in “c_Code Source Files”.

A.1.1 Executable Files User Manual

In order to perform a correct use of the source code, the user should do a few tasks before usage. Please have now a look at the folder “Executable File”:



Name	Date modified	Type	Size
CastaliaSPH	19/06/2014 13:32	File	22 KB
EGM96	18/06/2014 23:24	File	3.024 KB
ITGGOCE	19/06/2014 0:07	File	1.341 KB
ITGGRACE	19/06/2014 0:07	File	753 KB
List_Bodies.txt	20/06/2014 19:58	TXT File	1 KB
SphCoeff_WGS84_Mesh_1E6_deg20	20/06/2014 19:34	File	11 KB
SphCoeff_WGS84_Mesh_5E3_deg20	20/06/2014 19:35	File	11 KB
SphCoeff_WGS84_Mesh_5E4_deg20	20/06/2014 19:35	File	11 KB
SphCoeff_WGS84_Mesh_8E3_deg20	20/06/2014 19:35	File	11 KB
TFG_C_exe	20/06/2014 20:04	Application	91 KB
TFG_C_x64.exe	20/06/2014 19:56	Application	106 KB
WGS84_Mesh_1E2.txt	20/06/2014 12:34	TXT File	26 KB
WGS84_Mesh_1E3.txt	20/06/2014 12:34	TXT File	264 KB
WGS84_Mesh_1E4.txt	17/05/2014 16:25	TXT File	2.696 KB

The software user must in first place open the file called ”List_Bodies.txt”. This file is opened by the program to read the required information about central bodies, the available spherical harmonics mass coefficients and the polyhedrons.

It is structured in the following way:

- New Body Definition:
 - Body Number Assignment (check no repetitions)
 - Body Name
 - Gravitational parameter in $\left[\frac{m^3}{s^2}\right]$
 - Reference sphere radius in [m]
 - Body mean density $\left[\frac{kg}{m^3}\right]$
 - Body rotational speed in [rads/sec]

Therefore assigning the Earth for the first time would left a line as:

1 Earth 3.9860044150e+14 6.3781363000e+06 5.513792251126140e+03 7.2921150e-5

- Spherical Harmonics Mass Coefficients:
 - first character always a “C” (from coefficients)
 - Body Number (making possible to relate the spherical harmonics mass coefficients information to a the characteristics of the body as defined previously)
 - The file name containing the spherical harmonics mass coefficients
 - The degree of the model

Therefore assigning the spherical harmonics obtained from EGM96 would be a line as:

C 1 EGM96 360

Note that this implies that it should exist a file called ”EGM96” in the directory containing the information about spherical harmonics mass coefficients of Earth. It is not needed to add a new central body for every new addition of spherical harmonics mass coefficients, that is why body numbers are used to link the coefficients to the bodies.

- Polyhedron data addition:
 - first character always a “D” (Discrete)
 - Body Number (making possible to relate the polyhedron information to a the characteristics of the body as defined previously)
 - The number of faces of the polyhedron
 - The file name containing the information of the polyhedron

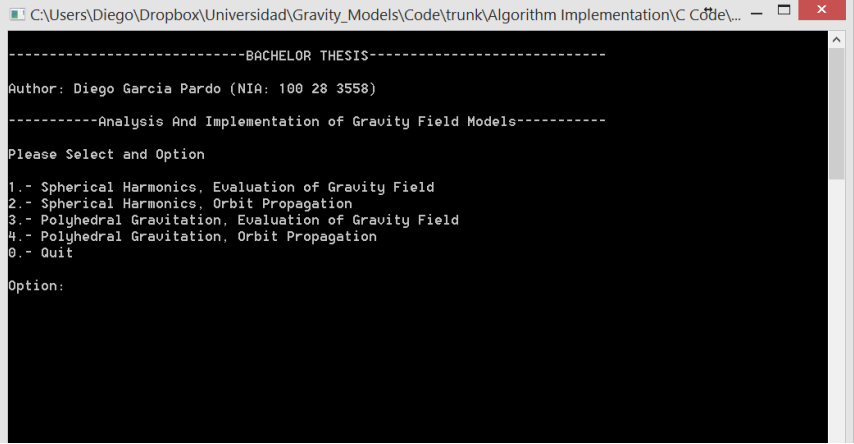
This assignation can be performed with line structured in this way:

D 3 1E2 WGS84_Mesh_1E2.txt

Once done this tasks, it is straight forward to make simple calculations with the executable.

A.1.1.1 Executable File, Simple Calculations

The usage of the executable file is straight forward and guided through the menu



```
-----BACHELOR THESIS-----  
Author: Diego Garcia Pardo (NIA: 100 28 3558)  
-----Analysis And Implementation of Gravity Field Models-----  
Please Select and Option  
1.- Spherical Harmonics, Evaluation of Gravity Field  
2.- Spherical Harmonics, Orbit Propagation  
3.- Polyhedral Gravitation, Evaluation of Gravity Field  
4.- Polyhedral Gravitation, Orbit Propagation  
0.- Quit  
Option:
```

At this point, I would like to indicate once again that this implementation uses as an integrator: 4th order fixed step but multiple step “Runge Kutta” following the guidelines found in [4] and [9]

The calculations performed in C for the computation of the gravity field using the algorithm described in Part II are faster than the MatLab implementation. However the difference can only be appreciated for long calculations mainly.

Something different happens in the calculations performed using the polyhedral algorithm described in Part III where the implementation in C is far faster than the MatLab implementation. This occurs since there exists numerous matricial multiplications where compiled languages take advantage over interpreted languages (as it is MatLab). The computation in C can be up to 35 times faster.

Chapter A.2

Implementation in MatLab

The MatLab implementation can be found in the folder adjacent to the one where all C files were found. This folder has been splitted in 3 different folders labelled with names Part II, Part III and Part IV being analogous to the file structure followed in this report. In this case there is no a "master script" from where to choose one calculation option or another.

A.2.1 Part II, Spherical Harmonics

In this folder it is possible to find several files that are required to perform the evaluation of the gravity field as well as orbit propagations using MatLab integrator. As explained in II ode113 seems to be the more efficient integrator (Implementation of the Adams-Basforth-Moulton multistep, variable step size and variable order integrator).

The file called "Acc_Field.m" computes the gravitational field in body fixed axis. This is a function called from the file "Field_Integrator.m" where non inertial terms are added. Examples of computation can be found in the file called "MainProgram.c" where an orbit propagation is done.

A.2.2 Part III, Discrete Body Gravitation

In this folder it is possible to find several files required for the evaluation of the gravitational field around a body. There exist two functions that are the core of the algorithm. These are:

- “PolyhedronParsev5.m”
- “Acceleration.m”

The first function recollects and obtains the data from the plain text source file where the polyhedral information is retained. This function requires a big amount of memory as it creates an auxiliary variable “Link_Check” that is used to compare if an edge has been already stored or not. Note that every edge is traversed twice because two adjacent faces share at least one edge, thus keeping the counter clockwise order, each edge is traversed opposite. For an example of usage open the file called: “Main.m” in this folder.

A.2.3 Part IV, Spherical Harmonics Mass Coefficients of a Constant Density Polyhedron

Opening the folder corresponding to this section, two different folders appear. Once is called “WGS84 Mesh Generator” and the other “Spherical Harmonics Derived From Polyhedron”.

In the first folder, with path:

Algorithm Implementation\MatLab Code\Part IV Spherical Harmonics Derived From Polyhedron\Spherical Harmonics Derived From Polyhedron\WGS84 Mesh Generator

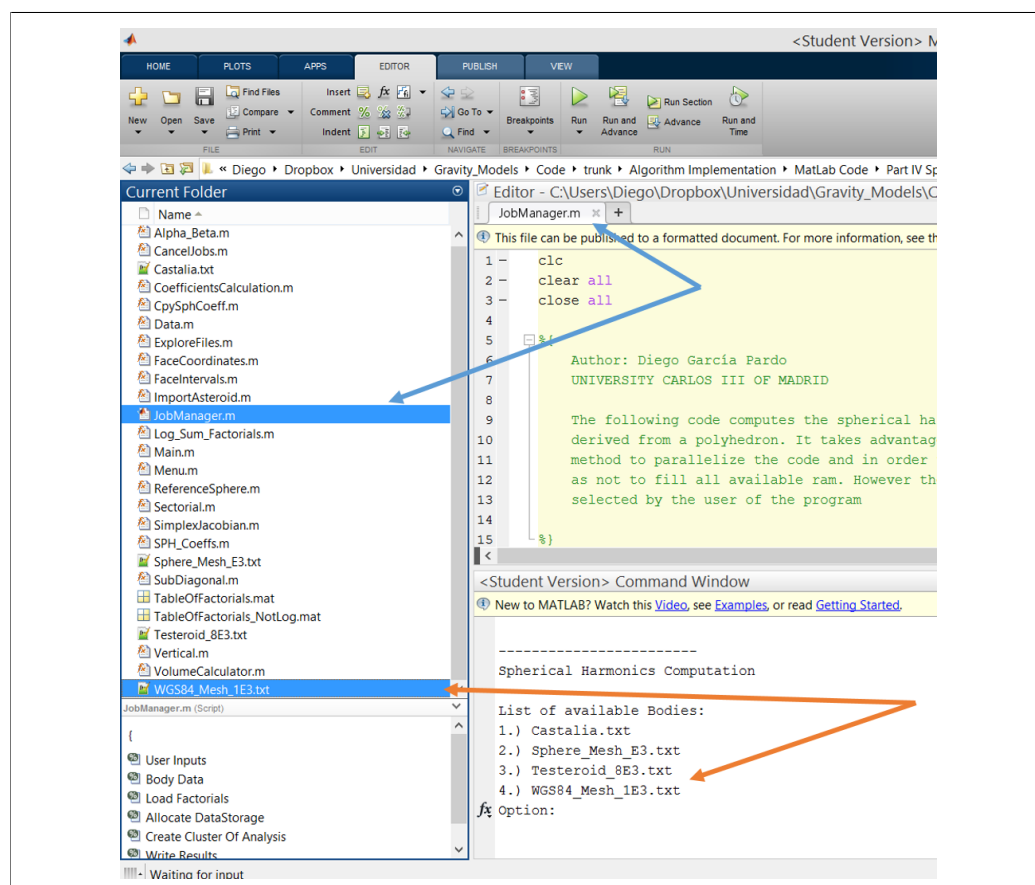
it is possible to find the scripts that allow to perform a correct discretization of the WGS84 ellipsoid keeping constant the original WGS84 ellipsoid volume

and to save the mesh to both “.mat” and “.txt” files to be used by the MatLab and C algorithm implementations.

In the second folder with path:

\Algorithm Implementation\MatLab Code\Part IV Spherical Harmonics Derived From Polyhedron\Spherical Harmonics Derived From Polyhedron\Parallel Based (CPU Based) V5

The script called “JobManager.m” performs the code parallelization as well as the require face splits in order to achieve better memory management. In order to perform the computation of the spherical harmonics mass coefficients from a polyhedron, the information containing the data from the discrete body (for example those files computed in the WGS84 Mesh generator) should be in the directory. The script finds all possible candidates in the path to evaluate the spherical harmonics.



After selecting the desired body, it is required to enter the degree and the desired number of face divisions. Recall now that the memory required for the computation grows **linearly with the number of faces** and **quadratically with the degree of the model**.

Once these parameters are set, the code performs the required calculations and once finished the results are saved to hard disk.

Part B

ANNEX, Economical Cost

Project Cost Summary	
Software Licensing	
MatLab Comercial License R2014a	2.000,00 €
Office Professional 2013	539,00 €
Visual Studio Professional 2013	677,00 €
Studio Rent	
Annual Rent	24.000,00 €
Annual Maintenance Cost	2.000,00 €
Personal Costs	
Junior Aerospace Engineer Annual Salary	35.000,00 €
Social Security Cost (Assuming 26,5%)	9.275,00 €
Total Project Cost	73.491,00 €

The previous information was found at:

- MatLab Commercial License pricing:
<http://www.mathworks.es/pricing-licensing/index.html?intendeduse=comm&prodcode=ML>
- Office Professional 2013 pricing:
http://www.microsoftstore.com/store/mseea/es_ES/cat/Office/categoryID.66226700
- Visual Studio Professional 2013 pricing:
http://www.amazon.es/Microsoft-Visual-Studio-Professional-2013/dp/B00GSVZXF2/ref=sr_1_3?ie=UTF8&qid=1403369495&sr=8-3&keywords=visual+studio+2013

- Studio Annual Rent:

<http://www.idealista.com/inmueble/25303248/>

- Maintenance Cost

<http://www.expansion.com/elmundo/2011/07/28/suvivienda/1311837711.html>

- Junior Aerospace Engineer Annual Salary:

http://www.prospects.ac.uk/aeronautical_engineer_salary.htm

- Social Security Cost

http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci1077Basesytiposdecotiza36537/index.htm

Bibliography

- [1] Robert A. Werner. Spherical Harmonic Coefficients For the Potential Of A Constant-Density Polyhedron. 1997.
- [2] Colombi Andrew C. *Quick Evaluation of Small Body Gravitation*. 2002.
- [3] Lance Benner. NASA Radar Imaging Of Near Earth Asteroids, 2013.
- [4] Mark S. Gockenbach. *Partial Differential Equations Analytical and Numerical Methods*. 2011.
- [5] Olivier Jamet and Emilie Thomas. A linear Algorithm for computing The Spherical Harmonics of the gravitational potential of a constant density polyhedron. 2004.
- [6] Oliver Montenbruck and Everhard Gill. *Satellite Orbits*. 2000.
- [7] Rummel R. *Geoid And Gravity in Earth Sciences, An Overview*. 2004.
- [8] A. Eckman Randy, J. Brown Aaron, and R. Adamo Daniel. Normalization of Gravitational Acceleration Models. 2011.
- [9] L.Burden Richard and Faires J. Douglas. *Numerical Analysis*. 2005.
- [10] D. A. Vallado. *Fundamentals of Astrodynamics and Applications*. 2007.
- [11] R. A. Werner and D. J. Scheeres. Exterior Gravitation of a Polyhedron Derived and Compared with Harmonic and Mascon Gravitation Representations of Asteroid 4769 Castalia. *Celestial Mechanics and Dynamical Astronomy*, 65:313–344, 1997.